Université de Montréal

Sequential Machine Learning Approaches for Portfolio Management

Nicolas Chapados
Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

Novembre 2009

About the cover page. The word mosaic on the cover page was obtained from the set of N-grams in this thesis ($1 \leq N \leq 3$), hand-filtered for relevance, and with size approximately proportional to frequency of occurrence. The layout was composed automatically using the Wordle internet utility (`www.wordle.net`), and final adjustments made with the Adobe Illustrator software.

Revision History
13 March 2009 • Original Version
26 November 2009 • Minor Corrections

Université de Montréal
Faculté des études supérieures

Cette thèse intitulée :

Sequential Machine Learning Approaches for Portfolio Management

présentée par :

Nicolas Chapados

a été évaluée par un jury constitué des personnes suivantes :

Pascal Vincent ,	président-rapporteur
Yoshua Bengio ,	directeur de recherche
Michel Gendreau ,	membre du jury
Doina Precup ,	examinateur externe
Yves Sprumont ,	représentant du doyen de la FÉS.

Thèse acceptée le 16 octobre 2009.

Summary

THIS THESIS CONSIDERS a number of approaches to make machine learning algorithms better suited to the sequential nature of financial portfolio management tasks.

We start by considering the problem of the general composition of learning algorithms that must handle temporal learning tasks, in particular that of creating and efficiently updating the training sets in a sequential simulation framework. We enumerate the desiderata that composition primitives should satisfy, and underscore the difficulty of rigorously and efficiently reaching them. We follow by introducing a set of algorithms that accomplish the desired objectives, presenting a case-study of a real-world complex learning system for financial decision-making that uses those techniques.

We then describe a general method to transform a non-Markovian sequential decision problem into a supervised learning problem using a K-best paths search algorithm. We consider an application in financial portfolio management where we train a learning algorithm to directly optimize a Sharpe Ratio (or other risk-averse non-additive) utility function. We illustrate the approach by demonstrating extensive experimental results using a neural network architecture specialized for portfolio management and compare against well-known alternatives.

Finally, we introduce a functional representation of time series which allows forecasts to be performed over an unspecified horizon with progressively-revealed information sets. By virtue of using Gaussian processes, a complete covariance matrix between forecasts at several time-steps is available. This information is put to use in an application to actively trade price spreads between commodity futures contracts. The approach delivers impressive out-of-sample risk-adjusted returns after transaction costs on a portfolio of 30 spreads.

Keywords: machine learning, portfolio management, artificial neural networks, Gaussian processes, approximate dynamic programming, non-additive utility optimization, time-series forecasting, commodity spreads.

Résumé

Cette thèse envisage un ensemble de méthodes permettant aux algorithmes d'apprentissage statistique de mieux traiter la nature séquentielle des problèmes de gestion de portefeuilles financiers.

Nous débutons par une considération du problème général de la composition d'algorithmes d'apprentissage devant gérer des tâches séquentielles, en particulier celui de la mise-à-jour efficace des ensembles d'apprentissage dans un cadre de validation séquentielle. Nous énumérons les desiderata que des primitives de composition doivent satisfaire, et faisons ressortir la difficulté de les atteindre de façon rigoureuse et efficace. Nous poursuivons en présentant un ensemble d'algorithmes qui atteignent ces objectifs et présentons une étude de cas d'un système complexe de prise de décision financière utilisant ces techniques.

Nous décrivons ensuite une méthode générale permettant de transformer un problème de décision séquentielle non-Markovien en un problème d'apprentissage supervisé en employant un algorithme de recherche basé sur les K meilleurs chemins. Nous traitons d'une application en gestion de portefeuille où nous entraînons un algorithme d'apprentissage à optimiser directement un ratio de Sharpe (ou autre critère non-additif incorporant une aversion au risque). Nous illustrons l'approche par une étude expérimentale approfondie, proposant une architecture de réseaux de neurones spécialisée à la gestion de portefeuille et la comparant à plusieurs alternatives.

Finalement, nous introduisons une représentation fonctionnelle de séries chronologiques permettant à des prévisions d'être effectuées sur un horizon variable, tout en utilisant un ensemble informationnel révélé de manière progressive. L'approche est basée sur l'utilisation des processus Gaussiens, lesquels fournissent une matrice de covariance complète entre tous les points pour lesquels une prévision est demandée. Cette information est utilisée à bon escient par un algorithme qui transige activement des écarts de cours (*price spreads*) entre des contrats à terme sur commodités. L'approche proposée produit, hors échantillon, un rendement ajusté pour le risque significatif, après frais de transactions, sur un portefeuille de 30 actifs.

Mots-clés: apprentissage machine, gestion de portefeuille, réseaux de neurones artificiels, processus Gaussiens, programmation dynamique approximative, optimisation de fonctions d'utilité non-additives, prévision de séries chronologiques, écarts de cours sur contrats à terme.

Contents

Summary . . . . . **v**

Résumé . . . . . **vii**

Contents . . . . . **ix**

List of Figures . . . . . **xv**

List of Tables . . . . . **xxi**

List of Listings . . . . . **xxv**

Acknowledgements . . . . . **xxvii**

1 Introduction **1**
1.1 Goals and Structure of this Thesis 5
1.2 Basic Definitions and Notation 6
1.2.1 Simple Returns 6
1.2.2 Risk-Free Asset 7
1.2.3 Other Conventions 7

2 Portfolio Choice **9**
2.1 Single-Period Problems 9
2.1.1 Basic Formulation 10
2.1.2 Solution 11
2.1.3 Risk-Free Asset, Tangency Portfolio, Separation . . . 12
2.1.4 Utility Maximization 14
2.1.5 Risk Measures 16
2.1.6 Additional Constraints 19
2.1.7 Forecasting Models 24
2.1.8 Forecast Stability and Econometric Issues 31
2.2 Multiperiod Problems 37
2.2.1 The Discrete-Time Case 38
2.2.2 The Continuous-Time Case 44
2.2.3 Structure of Optimal Solutions 47
2.2.4 The Martingale Formulation 49
2.2.5 Investor Learning 55

2.2.6 Common Extensions . . . 55
2.3 Direct and Alternative Methods for Portfolio Choice . . . 59
2.3.1 Machine Learning Approaches . . . 60
2.3.2 Parametric Portfolio Policies . . . 61
2.3.3 Nonparametric Portfolio Weights . . . 62
2.3.4 "Non-Allocation" Approaches . . . 63
2.3.5 Information-Theoretic Approaches . . . 64
2.3.6 Stochastic Programming Approaches . . . 65

3 Machine Learning and Approximate Dynamic Programming 69
3.1 Useful Definitions in Machine Learning . . . 69
3.1.1 Supervised Learning . . . 69
3.1.2 Function Classes . . . 71
3.1.3 The Bias–Variance Trade-Off . . . 73
3.2 Approximate Dynamic Programming . . . 77
3.2.1 Classical Approximation Methods . . . 77
3.2.2 Types of Approximation . . . 80
3.2.3 Linear Approximators . . . 81
3.2.4 Direct Policy Learning and Actor-Critic Methods . . . 86

4 Training Graphs of Learning Modules for Sequential Data 89
4.1 Challenges with Sequential Validation . . . 91
4.1.1 Goals and Organization of this Chapter . . . 93
4.2 Correctness Issues in Sequential Learning . . . 94
4.2.1 Economic Data . . . 95
4.2.2 Vintage Data and Database Management Issues . . . 97
4.2.3 Vintage Data and Learning Algorithm Issues . . . 97
4.3 Learning Framework and Notation . . . 98
4.3.1 Learning . . . 98
4.3.2 Temporal Learning Networks . . . 104
4.4 Algorithms . . . 105
4.4.1 Variables and Notation . . . 105
4.4.2 Sequential Validation . . . 105
4.4.3 Training . . . 107
4.4.4 Output Computation . . . 109
4.4.5 Refinements and Limitations . . . 109
4.5 Domain Analysis . . . 110
4.5.1 Hierarchical Designs . . . 111
4.5.2 Network Designs . . . 111
4.5.3 Why Separate Training and Output Computation? . . . 113
4.6 Case Study . . . 114
4.6.1 Existing Systems . . . 114
4.6.2 Making Use of the Composition Primitives . . . 114

4.7 Discussion . . . 120

5 K-Best Paths Methods for Portfolio Management . . . 121
5.1 On Optimizing Sharpe Ratios . . . 123
5.1.1 Regularization . . . 125
5.1.2 Leverage Scale . . . 127
5.1.3 Optimal Policies . . . 128
5.1.4 Difficulty of Optimizing Realized Sharpe Ratios by Gradient Descent . . . 130
5.2 Problem Formulation . . . 133
5.2.1 Solving for a General Utility . . . 134
5.2.2 Generating Good Trajectories . . . 135
5.2.3 Known Uses . . . 135
5.2.4 Relationship to Approximate Dynamic Programming Methods . . . 136
5.2.5 Limitations . . . 137
5.2.6 Relationship with POMDPs . . . 137
5.3 Enumerating the K Best Paths . . . 138
5.3.1 Generalized Bellman's Equations . . . 139
5.3.2 Recursive Enumeration Algorithm and Data Structures 141
5.4 Application to Portfolio Optimization . . . 142
5.4.1 The `MinCost` Source Utility . . . 144
5.4.2 Target Utilities . . . 145
5.4.3 Choosing a Good K . . . 146
5.4.4 Incremental Sharpe Ratio Source Utility . . . 146
5.4.5 Making the Search More Efficient . . . 150
5.4.6 Optimization by K-Best Paths . . . 154
5.5 Ordinal Regression for Portfolio Allocation . . . 161
5.5.1 Ordinal Regression with Proportional Odds . . . 163
5.5.2 Architecture . . . 164
5.5.3 Network Training and Regularization Issues . . . 165
5.6 Experimental Questions and Methodology . . . 169
5.6.1 Controller Architectures . . . 169
5.6.2 Experimental Plan . . . 171
5.7 Experimental Results . . . 173
5.7.1 Summary of Detailed Results . . . 173
5.7.2 Country-Specific Financial Performance . . . 177
5.7.3 Sharpe Ratio Difference Significance Tests . . . 178
5.7.4 Pooling Performance Results . . . 186
5.8 Discussion and Future Work . . . 189
5.A Appendix: Statistical Techniques . . . 192
5.A.1 Bootstrap Inference for the Sharpe Ratio . . . 192
5.A.2 Analysis of Variance . . . 194
5.A.3 Tukey's Adjustment for Multiple Comparisons . . . 196
5.B Appendix: Input Variables . . . 197

5.B.1 Technical . . . 197
5.B.2 Notes on Time-Series Regressions . . . 198
5.B.3 United States . . . 199
5.B.4 Canada . . . 202
5.B.5 Europe . . . 204
5.C Appendix: Detailed U.S. Model Results . . . 212
5.C.1 Linear Regression . . . 213
5.C.2 Gaussian Process . . . 213
5.C.3 Classification Neural Network . . . 215
5.C.4 Financial Neural Network . . . 217
5.C.5 Overall Results . . . 220
5.D Appendix: Detailed Canadian Model Results . . . 223
5.D.1 Linear Regression . . . 223
5.D.2 Gaussian Process . . . 225
5.D.3 Classification Neural Network . . . 225
5.D.4 Financial Neural Network . . . 227
5.D.5 Overall Results . . . 230
5.E Appendix: Detailed European Model Results . . . 233
5.E.1 Linear Regression . . . 233
5.E.2 Gaussian Process . . . 234
5.E.3 Classification Neural Network . . . 235
5.E.4 Financial Neural Network . . . 237
5.E.5 Overall Results . . . 240

6 Augmented Functional Time Series Representation and Forecasting with Gaussian Processes . . . 243
6.1 On Commodity Spreads . . . 246
6.2 Review of Gaussian Processes for Regression . . . 247
6.2.1 Basic Concepts for the Regression Case . . . 247
6.2.2 Choice of Covariance Function . . . 250
6.2.3 Optimization of Hyperparameters . . . 251
6.2.4 Two-Step Training . . . 252
6.2.5 Parameterization for Positive Hyperparameters . . . 253
6.2.6 Weighting the Importance of Input Variables: Automatic Relevance Determination . . . 253
6.3 Forecasting Methodology . . . 254
6.3.1 Functional Representation for Forecasting . . . 255
6.3.2 Augmented Functional Representation . . . 257
6.3.3 Input and Target Variable Preprocessing . . . 259
6.3.4 Training Set Subsampling . . . 260
6.3.5 Covariance Function Engineering . . . 260
6.4 Experimental Setting . . . 261
6.4.1 Methodology . . . 262
6.4.2 Models Compared . . . 262
6.4.3 Significance of Forecasting Performance Differences . . . 265

6.5 Forecasting Performance Results . . . 268
6.6 From Forecasts to Trading Decisions . . . 268
6.7 Financial Performance . . . 271
6.8 Discussion . . . 280
6.9 Appendix: Analysis of Forecasting Errors for the Wheat 3–7 Spread . . . 281

7 Conclusions and Summary of Contributions . . . 289
7.1 Summary of Contributions . . . 289
7.1.1 Chapter 2: Portfolio Choice . . . 289
7.1.2 Chapter 4: Training Graphs of Learning Modules for Sequential Data . . . 289
7.1.3 Chapter 5: K-Best Paths Methods for Portfolio Management . . . 290
7.1.4 Chapter 6: Augmented Functional Time Series Representation and Forecasting with Gaussian Processes . 291
7.2 Outlook for Future Research . . . 292

A Appendix . . . 295
A.1 Minimization of a Quadratic Form Under Linear Equality Constraints . . . 295
A.2 Deriving the Tangency Portfolio . . . 296
A.2.1 Efficient Frontier . . . 297
A.2.2 Maximization of the Sharpe Ratio . . . 297
A.3 Itô's Lemma . . . 298
A.3.1 Wiener Processes . . . 298
A.3.2 One-Dimensional Case . . . 298
A.3.3 Multi-Dimensional Case . . . 299
A.4 Inference for the Normal Distribution . . . 299
A.4.1 Marginalization and Conditioning . . . 300
A.4.2 Application to Gaussian Processes . . . 302

Glossary . . . 303

References . . . 305

Author Index . . . 333

Subject Index . . . 339

List of Figures

1.1 Trade-off between risk and return in "modern portfolio theory". 2
1.2 Illustration of the time conventions followed in this document. 7

2.1 Methodological steps surrounding the Markowitz single-period investment process. 10
2.2 Efficient frontier obtained from four assets. 13
2.3 90% Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) for a Student $t(3)$ distribution. 18
2.4 Fraction of wealth invested in the risky asset for a two-asset problem. 43
2.5 Optimal policy and value function as a function of time-to-maturity and initial dividend yield. 43
2.6 Terminal utility function for the stochastic programming asset allocation example. 66

3.1 Illustration of a feed-forward neural network. 74
3.2 Decomposition of the generalization error into bias and variance. 75
3.3 Bias–Variance trade-off curve. 76

4.1 Illustration of sequential validation. 91
4.2 Elementary composition of function approximators. 92
4.3 Example of complex composition of learning algorithms required by a financial decision-making application. 94
4.4 Illustration of forecasting at horizon h. 95
4.5 Example of *post hoc* revisions to macroeconomic time-series. . 96
4.6 Illustration of the framework that is assumed for temporal learning. 99
4.7 Generative view of the Kalman filter as a probabilistic graphical model. 101
4.8 Example of a directed acyclic graph of composed learning modules and "dirty" status propagation. 108
4.9 Functional blocks of a financial portfolio allocation system. . 115
4.10 Details of the technical model. 116
4.11 Details of the fundamental model. 117
4.12 Cumulative return of the composite-learner-based strategy. . 119

5.1 Illustration of the curse of dimensionality. 122

5.2 Two-period empirical Sharpe ratio as a function of asset weights $\mathbf{w}_1$ and $\mathbf{w}_2$, computed from 250 simulated trajectories of a VAR process. . . . 126
5.3 Divergences in the empirical Sharpe ratio remain at longer horizons. . . . 127
5.4 Leverage scale specification in Sharpe-like optimization criteria. 128
5.5 Fraction of wealth initially invested in the risky asset for the two-asset problem under the regularized Sharpe ratio objective. 129
5.6 Optimal policy under the Sharpe ratio utility for the third time-step of a ten time-step problem, given the previous two time-step realizations. . . . 130
5.7 Variance in Sharpe ratios optimized by conjugate gradient descent as transaction costs are varied. . . . 132
5.8 Distribution of the standard deviation of optimized Sharpe ratios. . . . 133
5.9 Summary of the proposed algorithm for finding good trajectories under a non-additive utility function. . . . 135
5.10 Intuition behind the recursive relationship underlying the REA K-best-paths algorithm. . . . 139
5.11 Data structure (fragment) for enumerating the K best paths ending at vertex t in a graph where vertices $u, v, w \in \Gamma^{-1}(t)$. 143
5.12 Formulation of the portfolio management problem as a shortest-path problem in a graph. . . . 143
5.13 Exploiting the correlation between source and target utilities: the number K of extracted paths should be large enough to adequately sample the "good target utility" region. . . . 147
5.14 Utility as a function of the extracted path index, in the order found by the K-best-paths algorithm. . . . 148
5.15 Kernel density estimate of the relationship between the incremental Sharpe ratio (source utility) and the Sharpe ratio (target utility) for a typical trajectory. . . . 149
5.16 Illustration of the beam search applied to the state-action graph. . . . 151
5.17 Efficient enumeration of the actions in order of decreasing dot product with the asset return vector. . . . 153
5.18 Impact of K, the number of extracted paths, on the Sharpe Ratio objective. . . . 156
5.19 Impact of source utility on Sharpe ratio after fine-tuning with gradient optimization, as a function of transaction costs. . . . 156
5.20 Plot of the difference in Sharpe ratio between decision sequences optimized by two different procedures. . . . 158
5.21 Mean Sharpe ratio after K-best paths rescoring. . . . 159
5.22 Sharpe ratio difference between the `IncrSharpe` and the `MinCost` source utility functions after K-best search but before gradient-based fine-tuning. . . . 160

5.23 Median difference between the `IncrSharpe` and `MinCost` source utilities. . . . 161
5.24 Schematic diagram of a Financial Neural Network. . . . 164
5.25 Average Sharpe ratio loss of using additional hidden units with the Classification Neural Network. . . . 176
5.26 United States: Cumulative return of the selected model of each type. . . . 181
5.27 United States: Annual returns of the selected model of each type. . . . 181
5.28 Canada: Cumulative return of the selected model of each type. 183
5.29 Canada: Annual returns of the selected model of each type. . 183
5.30 Europe: Cumulative return of the selected model of each type. 185
5.31 Europe: Annual returns of the selected model of each type. . 185
5.32 Q-Q plot of the residuals of the linear mixed-effects model against a theoretical normal distribution. . . . 189
5.33 Graphical result of Tukey's honestly significant differences procedure . . . 196
5.34 Hyperparameter comparison for Linear Regression models. United States, 1990–1998. . . . 214
5.35 Hyperparameter comparison for Linear Regression models. United States, 1999–2007. . . . 214
5.36 Hyperparameter comparison for Gaussian Process models. United States, 1990–1998. . . . 215
5.37 Hyperparameter comparison for Gaussian Process models. United States, 1999–2007. . . . 216
5.38 Hyperparameter comparison for Classification Neural Networks models. United States, 1990–1998. . . . 217
5.39 Hyperparameter comparison for Classification Neural Networks models. United States, 1999–2007. . . . 218
5.40 Hyperparameter comparison for Financial Neural Networks models. United States, 1990–1998. . . . 220
5.41 Hyperparameter comparison for Financial Neural Networks models. United States, 1999–2007. . . . 221
5.42 Overall results, United States, 1990–1998. . . . 222
5.43 Overall results, United States, 1999–2007. . . . 222
5.44 Hyperparameter comparison for Linear Regression models. Canada, 1991–1998. . . . 224
5.45 Hyperparameter comparison for Linear Regression models. Canada, 1999–2007. . . . 224
5.46 Hyperparameter comparison for Gaussian Process models. Canada, 1991–1998. . . . 226
5.47 Hyperparameter comparison for Gaussian Process models. Canada, 1999–2007. . . . 226
5.48 Hyperparameter comparison for Classification Neural Networks models. Canada, 1991–1998. . . . 228

5.49 Hyperparameter comparison for Classification Neural Networks models. Canada, 1999–2007. 228
5.50 Hyperparameter comparison for Financial Neural Networks models. Canada, 1991–1998. 231
5.51 Hyperparameter comparison for Financial Neural Networks models. Canada, 1999–2007. 231
5.52 Overall results, Canada, 1991–1998. 232
5.53 Overall results, Canada, 1999–2007. 232
5.54 Hyperparameter comparison for Linear Regression models. Europe, 1991–1998. 234
5.55 Hyperparameter comparison for Linear Regression models. Europe, 1999–2007. 235
5.56 Hyperparameter comparison for Gaussian Process models. Europe, 1991–1998. 236
5.57 Hyperparameter comparison for Gaussian Process models. Europe, 1999–2007. 236
5.58 Hyperparameter comparison for Classification Neural Networks models. Europe, 1991–1998. 238
5.59 Hyperparameter comparison for Classification Neural Networks models. Europe, 1999–2007. 238
5.60 Hyperparameter comparison for Financial Neural Networks models. Europe, 1991–1998. 241
5.61 Hyperparameter comparison for Financial Neural Networks models. Europe, 1999–2007. 241
5.62 Overall results, Europe, 1991–1998. 242
5.63 Overall results, Europe, 1999–2007. 242

6.1 Random functions drawn from the Gaussian Process prior. . . 250
6.2 Overfitting phenomenon observed during two-stage Gaussian Process training. 253
6.3 Illustration of the regression variables around which the forecasting problem is specified. 256
6.4 Training set for March–July Wheat spread, containing data until 1995/05/19. 258
6.5 The augmented functional representation attempts to capture the relationship between the price at the target time $t_0 + \Delta$ and information available at the operation time t_0. 258
6.6 Forecasts made from several operation times, given the same training set. 259
6.7 Illustration of multiple forecasts, repeated every 25 days, of the 1996 March–July Wheat spread. 263
6.8 Illustration of sequential validation with strongly overlapping test sets. 266
6.9 Overview of the Cross-Correlation-Corrected Diebold-Mariano test results for the grain spreads. 270

6.10 Computation of the Information Ratio between each potential entry and exit points, given the spread trajectory forecast. . . 272
6.11 Return after transaction costs of a portfolio of 30 spreads traded according to the maximum information-ratio criterion. 273
6.12 Standardized forecasting squared error as a function of the forecast horizon on the March–July Wheat spread. 282
6.13 Standardized forecasting negative log likelihood as a function of the forecast horizon on the March–July Wheat spread. . . 283
6.14 Squared error difference between AugRQ/all-inp and AR1. . 284
6.15 NLL difference between AugRQ/all-inp and AR1. 284
6.16 Squared error difference between AugRQ/all-inp and AugRQ/less-inp. 285
6.17 NLL difference between AugRQ/all-inp and AugRQ/less-inp. 285
6.18 Squared error difference between AugRQ/all-inp and AugRQ/no-inp. 286
6.19 NLL difference between AugRQ/all-inp and AugRQ/no-inp. . 286
6.20 Squared error difference between AugRQ/all-inp and Linear/all-inp. 287
6.21 NLL difference between AugRQ/all-inp and Linear/all-inp. . 287
6.22 Squared error difference between AugRQ/all-inp and StdRQ/no-inp. 288
6.23 NLL difference between AugRQ/all-inp and StdRQ/no-inp. . 288

List of Tables

5.1 Summary statistics for the three markets covered in the experiments. . . . 172
5.2 Model choice stability. . . . 174
5.3 Stability of significant hyperparameters for each model. . . . 175
5.4 Summary of financial performance measures. . . . 179
5.5 Financial Performance for the United States. . . . 180
5.6 Annual Returns for the United States. . . . 180
5.7 Financial Performance for Canada. . . . 182
5.8 Annual Returns for Canada. . . . 182
5.9 Financial Performance for Europe. . . . 184
5.10 Annual Returns for Europe. . . . 184
5.11 Pairwise Sharpe ratio difference between all pairs of models (United States). . . . 187
5.12 Pairwise Sharpe ratio difference between all pairs of models (Canada). . . . 187
5.13 Pairwise Sharpe ratio difference between all pairs of models (Europe). . . . 187
5.14 Number of times that each model is "beaten" by another in the Sharpe ratio difference tables. . . . 188
5.15 Diagnostics from fitting a mixed-effects model for analyzing the performance of various model architectures. . . . 188
5.15 List of technical variables. . . . 197
5.16 List of fundamental variables for the United States. . . . 200
5.17 Transformed variables used for the United States (monthly model). . . . 200
5.19 Linear regression of monthly S&P 500 returns using U.S. input variables. . . . 201
5.20 List of fundamental variables for Canada. . . . 202
5.20 Linear regression of daily S&P 500 returns using U.S. input variables. . . . 203
5.21 Transformed variables used for Canada (monthly model). . . . 204
5.23 Linear regression of monthly TSX-Composite returns using Canadian input variables. . . . 205
5.24 Linear regression of daily TSX-Composite returns using Canadian input variables. . . . 206
5.24 List of fundamental variables for Europe. . . . 207
5.25 Transformed variables used for Europe (monthly model). . . . 207

5.27 Linear regression of monthly Eurostoxx returns using European input variables. . . . 209
5.28 Linear regression of daily Eurostoxx returns using European input variables. . . . 210
5.29 Hyperparameters associated with each model, as studied in the experimental results. . . . 212
5.30 ANOVA results for United States, Linear Regression model, Period ending in 1998. . . . 213
5.31 ANOVA results for United States, Linear Regression model, Period ending in 2007. . . . 213
5.32 ANOVA results for United States, Gaussian Process model, Period ending in 1998. . . . 214
5.33 ANOVA results for United States, Gaussian Process model, Period ending in 2007. . . . 215
5.34 ANOVA results for United States, Classification Neural Network model, Period ending in 1998. . . . 216
5.35 ANOVA results for United States, Classification Neural Network model, Period ending in 2007. . . . 216
5.36 ANOVA results for United States, Financial Neural Network (with K-best targets), Period ending in 1998, after isolating all interactions. . . . 219
5.37 ANOVA results for United States, Financial Neural Network (with K-best targets), Period ending in 2007, after isolating all interactions. . . . 219
5.38 ANOVA results for United States, Financial Neural Network (without K-best targets), Period ending in 1998. . . . 219
5.39 ANOVA results for United States, Financial Neural Network (without K-best targets), Period ending in 2007. . . . 220
5.40 ANOVA results for Canada, Linear Regression model, Period ending in 1998. . . . 223
5.41 ANOVA results for Canada, Linear Regression model, Period ending in 2007. . . . 223
5.42 ANOVA results for Canada, Gaussian Process model, Period ending in 1998. . . . 225
5.43 ANOVA results for Canada, Gaussian Process model, Period ending in 2007. . . . 225
5.44 ANOVA results for Canada, Classification Neural Network model, Period ending in 1998. . . . 227
5.45 ANOVA results for Canada, Classification Neural Network model, Period ending in 2007. . . . 227
5.46 ANOVA results for Canada, Financial Neural Network model (with K-best targets), Period ending in 1998. . . . 229
5.47 ANOVA results for Canada, Financial Neural Network model (with K-best targets), Period ending in 2007. . . . 229

5.48 ANOVA results for Canada, Financial Neural Network model (without K-best targets), Period ending in 1998. 230
5.49 ANOVA results for Canada, Financial Neural Network model (without K-best targets), Period ending in 2007. 230
5.50 ANOVA results for Europe, Linear Regression model, Period ending in 1998. 233
5.51 ANOVA results for Europe, Linear Regression model, Period ending in 2007. 234
5.52 ANOVA results for Europe, Gaussian Process model, Period ending in 1998. 235
5.53 ANOVA results for Europe, Gaussian Process model, Period ending in 2007. 235
5.54 ANOVA results for Europe, Classification Neural Network model, Period ending in 1998. 237
5.55 ANOVA results for Europe, Classification Neural Network model, Period ending in 2007. 237
5.56 ANOVA results for Europe, Financial Neural Network model (with K-best targets), Period ending in 1998. 239
5.57 ANOVA results for Europe, Financial Neural Network model (with K-best targets), Period ending in 2007. 240
5.58 ANOVA results for Europe, Financial Neural Network model (without K-best targets), Period ending in 1998. 240
5.59 ANOVA results for Europe, Financial Neural Network model (without K-best targets), Period ending in 2007. 240

6.1 Forecast performance difference between AugRQ/all-inp and all other models. 269
6.2 List of spreads used to construct the tested portfolio. 272
6.3 Performance on the 1 Jan. 1994–30 April 2007 period. 274
6.4 Performance on the 1 January 1994–31 December 2002 period. 275
6.5 Performance on the 1 January 2003–30 April 2007 period. . . 276
6.6 Yearly returns of the entire portfolio and of sub-portfolios formed by spreads on the same underlying commodity. 277
6.7 Correlation matrix of monthly returns among sub-portfolios formed by spreads on the same underlying commodity. 277
6.8 Financial performance of the $AR(1)$ model on the 30-spread portfolio. 278
6.9 Financial performance of the **Linear/all-inp** model on the 30-spread portfolio. 279

List of Listings

4.1 Sequential Validation 106
4.2 Training 107
4.3 Training Output Computation 108
4.4 Output Computation 109
4.5 State Transition 109
5.1 Recursive Enumeration Algorithm 141
5.2 Computing the Next Path 141
5.3 Dot-Product Order Enumeration Algorithm 153

Acknowledgements

This thesis could not have existed without the direct and indirect contributions of numerous individuals.

Above all, my deepest gratitude goes to my thesis advisor, mentor, business partner and friend, Yoshua Bengio. His tireless support, advice and insight helped me wander on profitable avenues. The combination of inspiration and freedom he imparts make it exceptional to pursue graduate studies under his supervision.

I would like to thank my colleagues at ApSTAT Technologies for helping me understand many of the practical difficulties of applying theoretical academic constructions: Réjean Ducharme, Charles Dugas, Christian Hudon, Xavier Saint-Mleux and Pascal Vincent. For many insightful conversations on machine learning, my fellow teammates in the LISA laboratory, in particular Aaron Courville and Hugo Larochelle. And to Éric-Paul Couture and Christian Dorion for their deep understanding of the practice and theory of financial markets.

I thank the (sadly now-dispersed) team at Desjardins Global Asset Management for the opportunity to immerse myself in a stimulating trading environment, in particular Robert Normand (now at Monexia), Jacques Lussier, Marc Boucher (now at Monexia), Éric Léveillé, Luc Veillette and Mariane Bastien (now at the STM Pension Fund).

For helping to greatly improve this thesis, I would like to thank the members of the jury for their comments on this work. I would also like to recognize the insightful suggestions of Éric-Paul Couture for some chapters.

For their financial support, I thank NSERC, FCAR and PRECARN.

For having made writing this thesis a slight bit less insufferable, I must express gratitude to Shirley Fan.

Finally, I warmly thank my parents, my father Camille for his advice and academic experience, and my mother Claire for never having doubted my capabilities.

À mes parents et
à mes petites sœurs.

1 Introduction

If economists could manage to get themselves thought of as humble, competent people, on a level with dentists, that would be splendid.
— *John Maynard Keynes*

Portfolio choice is a central problem of economic agents. In few words, it asks how one should spread one's wealth across a number of different assets. Of course, each asset is unique and offers its own outcome perspectives. These can be roughly summarized by an "expected return" and a "risk" aspect: the first one quantifies what would be the likely price appreciation of the asset or income arising from it over a given time period; the second measures how uncertain these payoffs to the investor can be.

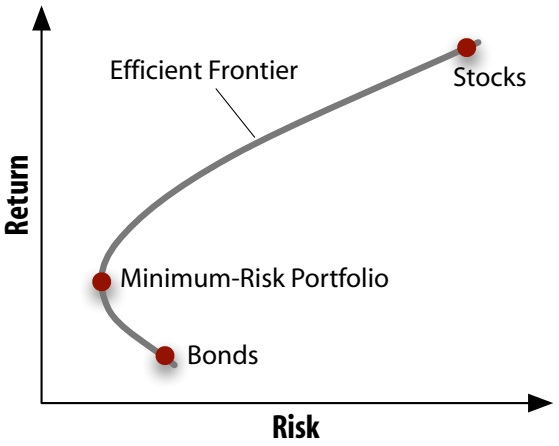
It has long been understood that there is a fundamental trade-off between these two aspects, yielding a continuum of opportunities: at one end of the spectrum, short-term government bonds provide very small returns with absolute certainty.* At the other end, small-cap growth stocks, for instance, may promise fabulous returns—but only if the company succeeds, for otherwise the investor may as well completely lose all her money.

Just as important is how individual risks combine at the portfolio level: what is the overall portfolio risk if assets are combined in specific ways? Real-world assets are not independent; some may zig as others zag. This is the fundamental idea behind the concept of *diversification*: the overall portfolio risk may be *less* than the sum of the risks of the individual assets that constitute it.

These ideas are summarized in Fig. 1.1, which illustrates two hypothetical individual assets, 'corporate bonds' and 'stocks', on the risk–return plane. These two assets are assumed to have well-defined risk and return characteristics. The figure also illustrates the notion of "efficient frontier", which traces out the risk and return characteristics of *portfolios* mixing the individual assets in specific proportions. The key insight of diversification appears in plain sight: there exists portfolios whose risk is lower than either asset, but with better return than the lowest-risk asset.

The first quantitative treatment of diversification in portfolios of assets is due to the seminal paper of Markowitz (1952), who introduced, among other concepts, the notion of the efficient frontier on the risk–return plane; the

*Assuming that the bond is denominated in the country's national currency.



◀ **Figure 1.1.** *Trade-off between risk and return in "modern portfolio theory". Portfolios on the efficient frontier yield the best return for a given risk level. Intermediate portfolios, mixing stocks and bonds, may exhibit lower risk than individual assets, a consequence of* diversification.

methodology introduced by Markowitz, and perfected ever since by countless others, has been called Modern Portfolio Theory (MPT). However, an intuitive understanding of the benefits of diversification came much earlier. As Rubinstein (2002), in his half-century retrospective of Markowitz's paper, observes,

> Markowitz was hardly the first to consider the desirability of diversification. Daniel Bernoulli in his famous 1738 article about the St. Petersburg Paradox argues by example that risk-averse investors will want to diversify: "... it is advisable to divide goods which are exposed to some small danger into several portions rather than to risk them all together" (Bernoulli 1738). As Markowitz (1999) himself points out in his historical review of portfolio theory, Bernoulli is also not the first to appreciate the benefits of diversification. For example, in *The Merchant of Venice*, Act I, Scene I, William Shakespeare has Antonio say:
>
> > ". . . I thank my fortune for it,
> > My ventures are not in one bottom trusted,
> > Nor to one place; nor is my whole estate
> > Upon the fortune of this present year . . ."
>
> Although this turns out to be a mistaken security, Antonio rests easy at the beginning of the play because he is diversified across ships, places, and time.

Until Markowitz (1952), the problem was approached on a "bottom-up" basis: each constituent (e.g. stock, bond) of the portfolio was chosen for its own risk and return characteristics, without regard for its interaction with

the rest of the portfolio.* However, due to diversification effects, this simple form of analysis is insufficient: the decision to hold a security should not only depend on a simple comparison of its expected risk and return profile to that of other securities, but also on its *marginal impact* on the risk–return profile of the investor's entire portfolio. Put differently, the decision to hold a security cannot be made in isolation, but is contingent upon the other securities that the investor already holds (or wants to hold). Earlier treatments of security analysis, including such classics as Graham and Dodd (1934) and Williams (1938), lack this perspective.

Myopia Dystopia

The original portfolio choice formulation by Markowitz has the investor make all her forecasts, of expected asset returns and covariances between them as we shall see in Chapter 2, at the start of an investment period, and then lets the investor rest until the end of the period. In particular, the investor is "prohibited" from tinkering with the allocations until the start of the next period. When that time comes, she acts as though any previous period never existed, or any further period will never exist: decisions are made strictly one period at a time. For this reason, Markowitz's formulation is called *single-period*.

It is also called "myopic", referring to the inability of the investor to see beyond the immediate future and anticipate future opportunities. Obviously, in practice, investors do not all die after one period, and a huge assortment of stratagems are employed to "repair" the single-period formulation to varying degrees and make it better reflect reality; Chapter 2 covers the most common ones.

However, even these fixes are insufficient since they do not reflect the fact that investment is, fundamentally, an extended process. The asset universe provides changing opportunities, some of which can be anticipated in advance. Perhaps the investor could want to consume a portion of her wealth along the way, or receives income from non-investment sources, changing the investable capital in known (or unknown) ways. Moreover, frictions abound in the process: there are costs to every trade, and governments are prompt to ask for a commission on any good deed (also known as "taxes"). Planning ahead for these contingencies, in fact for the complete future set of contingencies weighted by their probabilities, requires a drastically different viewpoint than that afforded by single-period approaches. They lead to the multiperiod formulations, first analyzed by Mossin (1968), Samuelson (1969) and Merton (1969) (see §2.2/p. 37).

A special group of investors commands specific requirements: that of *institutional investors*, in particular mutual or hedge fund managers operating in a competitive environment. Their main characteristic is that they are not

*Variance had been considered as a measure of financial risk as early as 1906 by Fisher (Fisher 1906).

only interested in maximizing the utility of their client's final wealth, but also in optimizing the trajectory that wealth takes to reach its final destination. Consider, for instance, a client choosing between two competitive funds offering similar returns; assuming that other fund characteristics are identical (including the stated investment risk profile), the client could well favor the fund having the "nicer" past return characteristics, where "nice" may not only include the variance of returns but more global criteria such as the *drawdown**.

DRAWDOWN: Worst decline suffered by an investment from its peak value.*

Furthermore, on a day-to-day basis, the fund manager does not only care about how he will perform at the end of a long horizon, but how he is performing *right now*. It is a tired Wall Street cliché to state that "you are only as good as your last market call." Tired, perhaps, but a plea for help from practitioners that has seemed relatively ignored by academics.

Regrettably, the traditional multiperiod formulations outlined previously turn out to be unsatisfactory for the demands of institutional fund management. As stated, a fund manager operating in a competitive market cares as much about the path as about the final outcome.[†] In other words, the realized performance picture must be as rosy as possible, for as much of the time as possible, because clients can choose to join and leave the fund on a fairly unrestricted basis.[‡] However, and this is a fatal mismatch, the utility functions assumed by the classical multiperiod solutions to the portfolio choice problem ignore these considerations and focus exclusively on the distribution of terminal wealth (generally in conjunction with an intermediate stream of consumption, which may be appropriate for a University endowment fund, but is irrelevant for a hedge fund or mutual fund manager). We argue that practitioners care about more dimensions of the picture than what has generally been assumed in the literature so far.

Role of Machine Learning

In this context, one may inquire what may statistical machine learning, sitting at a fruitful intersection between statistics and computer science (Bishop 2006), bring to the practice of portfolio choice. The answer is mainly twofold. First, by readily tackling the vast amounts of historical data that are available about financial markets, machine learning can provide insight into the *forecasting problem* at the core of traditional portfolio choice formulations, both single- and multiperiod ones. Second, it provides a framework for allowing to completely bypass forecasting issues altogether and make direct allocation choices. For instance, many multiperiod portfo-

[†]Other institutional investors, such as those working for defined-benefit pension funds, insurance companies, foundations and endowments are generally not subject to such stringent constraints.

[‡]Although the financial panic of the Fall of 2008 has made long fund lock-up periods fashionable again, the trend until that point had been for lock-ups to become shorter in the competitive hedge fund industry, several funds offering redemptions with a 30-day notice or less.

lio management problems are formulated in terms of dynamic programming, which, due to the *curse of dimensionality* (see Fig. 5.1, p. 122), restricts its applicability to fairly small problems. In the past fifteen years, a lot of theoretical and practical progress has been accomplished on methods to approximately solve dynamic programming problems, prominently with the help of approaches developed within the machine learning community.

This thesis investigates both of these threads. However, a principal difficulty—and objection—has been and remains in following a rigorous methodology: since history provides only a single realized trajectory, making repeated use of the same historical data leads to *retrospective bias*, excessively optimistic estimates of past performance that vanish when a system is put in deployment.* A well-known way of obtaining realistic estimates of performance is to follow a *sequential validation* (also known as *simulated out-of-sample* or *prequential*) methodology, which attempts to reproduce as closely as feasible the steps that would take a real-life decision-maker acting in real time. Nevertheless, this is easier said than done: practical portfolio allocation systems can be made up of complex networks of dozens of learning modules, whose interdependencies make it difficult to remain both rigorous and efficient. We pay due attention to these issues, since they are a necessity for any practical implementation.

1.1 Goals and Structure of this Thesis

This thesis considers a number of approaches to make machine learning algorithms better suited to the sequential nature of financial portfolio management tasks. It considers both models that are specialized for variants of the portfolio choice problem, and novel uses of classical models in a portfolio management context.

We set the stage, in Chapter 2, by providing an in-depth review of the classical theory of portfolio choice, covering numerous variants of the single-period approach, and both discrete-time and continuous-time multiperiod formulations. We also examine a number of alternative and "direct" methods that deviate from the orthodox theory.

We then briefly summarize (Chapter 3) important concepts from statistical learning algorithms and approximate dynamic programming. Our goal is not to provide an exhaustive survey of the literature but simply to recall important concepts for understanding this thesis.

As hinted at above, before we can propose improvements to existing practice, we must be able to get accurate simulated performance in a sequential validation context. We consider (Chapter 4) the problem of the general composition of learning algorithms that must handle temporal learning tasks,

*This problem is so endemic in finance that it is discounted into the expectations of portfolio managers, who generally take pure backtest results of a new model—one that does not have a real trading track record—with a sizable grain of salt.

in particular that of creating and efficiently updating the training sets in a sequential simulation framework. We enumerate the desiderata that composition primitives should satisfy, and underscore the difficulty of rigorously and efficiently reaching them. We follow by introducing a set of algorithms that accomplish the desired objectives, presenting a case-study of a real-world complex learning system for financial decision-making that uses those techniques.

In Chapter 5, we revisit the classical Samuelson–Merton paradigm of multiperiod optimal portfolio choice and examine how it can benefit from advances in the computation of K-shortest-paths algorithms to optimize non-time-separable utility functions that can be of particular relevance to institutional portfolio managers. We train a learning algorithm to directly optimize a Sharpe Ratio (or other risk-averse non-additive) utility function. We illustrate the approach by demonstrating extensive experimental results using a neural network architecture specialized for portfolio management and compare against well-known alternatives.

In Chapter 6, we introduce a functional representation of time series which allows forecasts to be performed over an unspecified horizon with progressively-revealed information sets. By virtue of using Gaussian processes, a complete covariance matrix between forecasts at several time-steps is available. This information is put to use in an application to actively trade price spreads between commodity futures contracts. The approach delivers impressive out-of-sample risk-adjusted returns after transaction costs on a portfolio of 30 spreads.

Finally, Chapter 7 concludes and summarizes the contributions of this work.

1.2 Basic Definitions and Notation

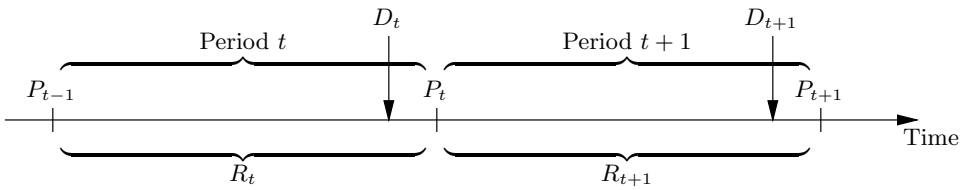
1.2.1 Simple Returns

In this thesis, we mostly consider the discrete-time scenario, in which *one period* (e.g. one day or one month) elapses between times t and $t+1$, where $t \in \mathbb{N}$. We define period t to be the one elapsed between times $t-1$ and t; see Figure 1.2.

Let $\{P_t\}, P_t \in \mathbb{R}_+$ be a random asset price process. We shall adopt the convention that any variable subscripted by a time index t can be measured given the set of information available at time t, which we denote $\mathcal{F}_t$.

Definition 1 *The* **simple rate of return** *of an asset during period t is given by*

$$R_t = \frac{P_t}{P_{t-1}} - 1.$$



◀ **Figure 1.2.** *Illustration of the time conventions followed in this document.*

For a dividend-paying asset, we consider dividends at time t, D_t, to be paid immediately before *recording price P_t. The simple return taking dividends into account is*

$$R_t = \frac{P_t + D_t}{P_{t-1}} - 1.$$

1.2.2 Risk-Free Asset

We denote by $R_{f,t}$ the rate of return earned by the risk-free asset (for instance, short-term government bonds).

1.2.3 Other Conventions

As much as possible, we attempt to adhere to the following notational conventions:

- Matrices and vectors are typeset in **bold face**; scalar variables are set in *italics*. The i-th element of vector $\mathbf{v}$ is $\mathbf{v}_i$; the i,j-th element of matrix $\mathbf{M}$ is $\mathbf{M}_{i,j}$. The i-th row of the matrix is $\mathbf{M}_{i,\cdot}$ and the j-th column is $\mathbf{M}_{\cdot,j}$.
- Matrix and vector transposition is indicated by a $'$ (prime).
- $\mathbf{M} \succ 0$ indicates that matrix $\mathbf{M}$ is positive-definite; $\mathbf{M} \succeq 0$ indicates that matrix $\mathbf{M}$ is semipositive-definite.
- It is sometimes useful to denote a vector of ones, whose size is appropriate given the context. We denote such a vector by the Greek letter iota, $\boldsymbol{\iota}$.

2 Portfolio Choice

Risk is a part of God's game, alike for men and nations.

— *Warren Buffett*

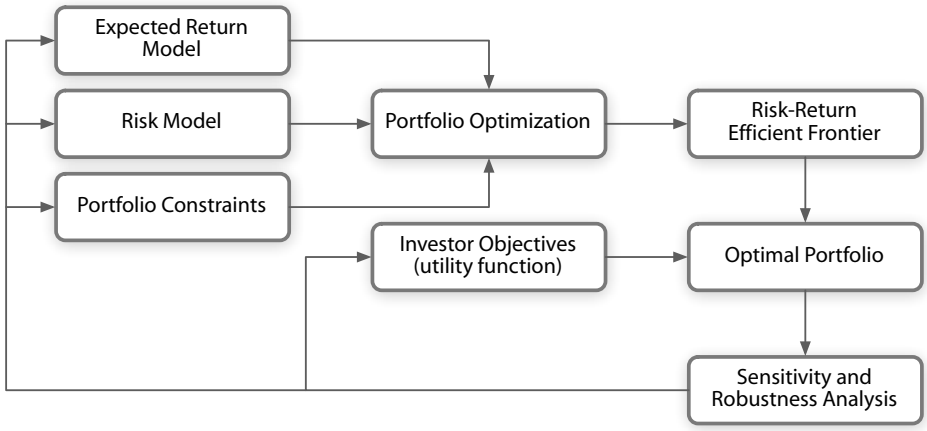
This chapter aims to review the main classical results about optimal portfolio construction, adopting a mostly-thematic rather than chronological perspective.

We start with the classical single-period "modern portfolio theory" of Markowitz (1952, 1959) and its numerous refinements (§2.1/p. 9), then proceed to the multiperiod and continuous-time formulations first studied by Mossin, Samuelson and Merton (§2.2/p. 37). A customary emphasis in this context has been to understand the structure of optimal solutions (under suitably analytically-tractable simplifications) and we shall examine the most enlightening of them.

Finally, straying from the traditional dynamic programming setting generally assumed in finance, we also examine various "direct" and alternative criteria for portfolio choice, mostly introduced in the machine learning and operations research communities (§2.3/p. 59).

2.1 Single-Period Problems

In the single-period portfolio choice problem, the investor is assumed to make allocation decisions once and for all at the beginning of a given period (e.g. one quarter or one year), based on estimated prospects for the risk and return relationships of a universe of N investable assets over the horizon. Once made, the allocation decisions are not allowed to change until the end of the period; the impact of decisions arising in subsequent periods is not considered in this case, and for this reason, single-period problems lead to so-called *myopic* policies. Markowitz (1952) introduced the basic formulation, including expressions for the expected portfolio return and variance in terms of the portfolio weights and expected returns, variances and covariances of individual assets. He also introduced the *efficient frontier* and its depiction on the mean-variance plane. Since the original formulation uses the asset variances (and covariances) as the risk measure, the methodology is often called *mean–variance allocation*.



▶ **Figure 2.1.** *Methodological steps surrounding the Markowitz single-period investment process; adapted from Exhibit 2.2 (p. 21) of Fabozzi, Focardi, and Kolm (2006).*

Despite their original conceptual simplicity, single-period problems are a large topic in which the optimization step is but one aspect. Just as important are the choice of utility function (§2.1.4/p. 14), risk measures (§2.1.5/p. 16), problem constraints (§2.1.6/p. 19) and forecasting models (§2.1.7/p. 24). Moreover, delicate issues related to the stability and econometrics of the obtained solutions need to be addressed for a successful implementation of the approach (§2.1.8/p. 31). This entails a rather involved methodology for single-period portfolio choice, which can be summarized by Fig. 2.1.

2.1.1 Basic Formulation

Let $\mathbf{R}_{t+1} \in \mathbb{R}^N$ be a vector of random *asset returns* between times t and $t+1$ (see §1.2/p. 6 for a summary of the time index conventions). Assume that the investor makes, given the information available at time t, a forecast of the first two moments of the distribution of future returns,

$$\begin{aligned} \boldsymbol{\mu}_{t+1|t} &= \mathbb{E}_t\left[\mathbf{R}_{t+1}\right] \\ \boldsymbol{\Sigma}_{t+1|t} &= \mathrm{Cov}_t\left[\mathbf{R}_{t+1}\right], \end{aligned}$$

where the $\mathbb{E}_t[\cdot]$ and $\mathrm{Cov}_t[\cdot]$ denote, respectively, the expectation and covariance matrix of a (vector) random variable conditioned on the information available at time t. For simplicity in this section, since single-period modeling does not explicitly consider the consequences of time, we drop the time subscripts on the above quantities, which we write simply as $\mathbf{R}$, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Likewise, the return on the risk-free asset during the period is denoted by R_f.

The investor allocates its capital among the N assets, forming a portfolio $\mathbf{w} \in \mathbb{R}^N$ where each element $\mathbf{w}_i$, the *weight* of asset i, represents the fraction

of total capital held in the asset. The expected portfolio return and variance are given respectively by

$$\mu_P = \mathbf{w}'\boldsymbol{\mu} \qquad \text{and} \qquad \sigma_P^2 = \mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w}. \tag{2.1}$$

We shall make the following assumptions about the assets:

1. There are no "redundant" assets, i.e. no asset return can be obtained as a linear combination of the returns of other assets.
2. All assets are risky (have positive return variance), which implies, in conjunction with the above assumption, that the covariance matrix $\boldsymbol{\Sigma}$ is nonsingular. (The inclusion of a risk-free asset is treated in §2.1.4/p. 14.)

Definition 2 (Efficiency) *A portfolio* $\mathbf{w}$ *is said to be* ***efficient*** *if it is the lowest-variance portfolio for a given level of expected return.*

The portfolio choice problem seeks to directly find efficient portfolios by determining an "optimal" vector of asset weights. The minimum-variance formulation of the problem considers the expected portfolio variance as the measure of risk. It takes the form

$$\begin{aligned} \mathbf{w}^* &= \arg\min_{\mathbf{w}} \frac{1}{2}\,\mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w} && (2.2)\\ \text{subject to} \quad \mathbf{w}'\boldsymbol{\mu} &= \rho, && (2.3)\\ \mathbf{w}'\boldsymbol{\iota} &= 1. && (2.4) \end{aligned}$$

The objective function, eq. (2.2), seeks the vector of weights which minimizes the total expected portfolio variance, subject to constraint (2.3) which requires a portfolio return of ρ (which can be viewed as the desired or target return), and constraint (2.4) which specifies that all capital must be invested. We consider other types of constraints — and their implication on the solution methods — in §2.1.6/p. 19.

2.1.2 Solution

Since all constraints are of equality type, problem (2.2) can be solved analytically by introducing Lagrange multipliers. The general solution is derived in §A.1/p. 295. To borrow notation from that section, we set

$$\mathbf{A} = \begin{pmatrix} \boldsymbol{\mu}' \\ \boldsymbol{\iota}' \end{pmatrix} \qquad\qquad \mathbf{b} = \begin{pmatrix} \rho \\ 1 \end{pmatrix},$$

and obtain the optimal weights $\mathbf{w}^*$ by substitution into eq. (A.10). Some algebraic manipulation yields the somewhat simplified but enlightening form (Merton 1972; Fabozzi, Kolm, Pachamanova, and Focardi 2007)

$$\mathbf{w}^* = \mathbf{g} + \mathbf{h}\rho, \tag{2.5}$$

where

$$\mathbf{g} = \frac{\boldsymbol{\Sigma}^{-1}(c\boldsymbol{\iota} - b\boldsymbol{\mu})}{d}, \qquad \mathbf{h} = \frac{\boldsymbol{\Sigma}^{-1}(a\boldsymbol{\mu} - b\boldsymbol{\iota})}{d},$$

and

$$a = \boldsymbol{\iota}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\iota}, \qquad b = \boldsymbol{\iota}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \qquad c = \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \qquad d = ac - b^2.$$

Similarly, the globally minimum-variance portfolio (GMV) is obtained without imposing the expected-return constraint, yielding portfolio weights and variance respectively given by

$$\mathbf{w}^*_{\text{GMV}} = \frac{\boldsymbol{\Sigma}^{-1}\boldsymbol{\iota}}{\boldsymbol{\iota}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\iota}} \qquad \text{and} \qquad \sigma^2_{\text{GMV}} = \frac{1}{\boldsymbol{\iota}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\iota}}. \tag{2.6}$$

The above solutions yield two important insights. First, as will be illustrated next, it reflects the benefits of diversification. Second, it highlights that ultimately, higher returns can only be obtained by taking on higher leverage — thence more risk — since the optimal weight vector is linear in the target return ρ.

To illustrate these solutions, consider a four-asset problem specified as

$$\boldsymbol{\mu} = \begin{bmatrix} 0.095 \\ 0.070 \\ 0.090 \\ 0.075 \end{bmatrix}, \qquad \boldsymbol{\Sigma} = \begin{bmatrix} 0.0380 & 0.0085 & 0.0089 & 0.0066 \\ 0.0085 & 0.0331 & 0.0156 & 0.0039 \\ 0.0089 & 0.0156 & 0.0334 & 0.0070 \\ 0.0066 & 0.0039 & 0.0070 & 0.0240 \end{bmatrix}.$$

The efficient frontier for this example is plotted in Fig. 2.2, under the label "Efficient Frontier (no risk-free asset)".

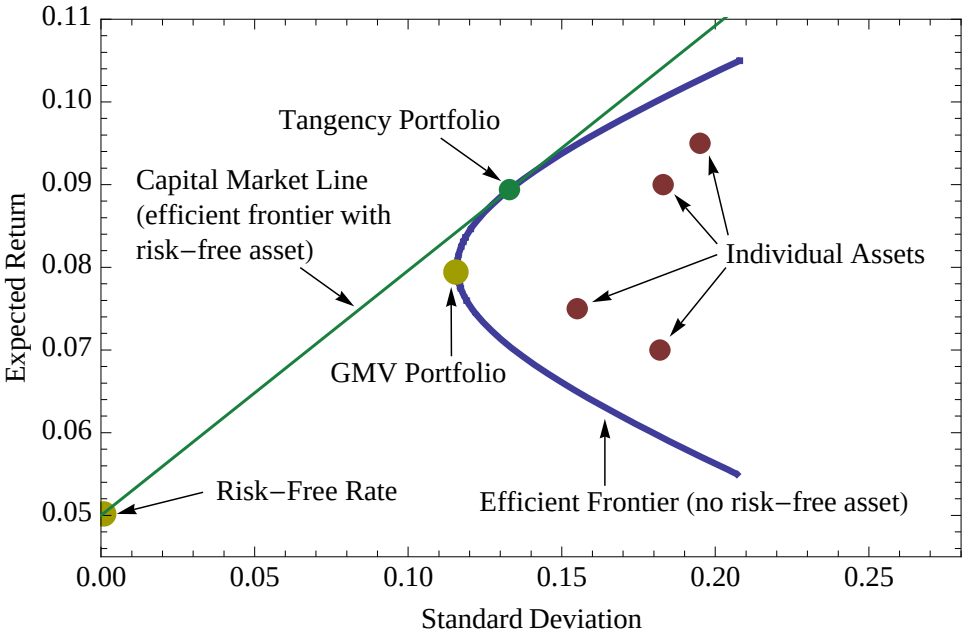
2.1.3 Risk-Free Asset, Tangency Portfolio, Separation

When one of the assets can be considered risk-free (i.e. a return variance of zero and necessarily an identically zero covariance with all other assets), the above formulation cannot be used directly since the covariance matrix $\boldsymbol{\Sigma}$ would not be invertible. In this context, it can be shown that all efficient portfolios are formed by a linear combination of the risk-free asset and the *tangency portfolio* located on the risky-assets efficient frontier. These portfolios are located on what is known as the Capital Market Line (CML). These concepts, for a risk-free rate of 5%, are depicted on Fig. 2.2.

As derived in §A.2/p. 296, the risky-asset proportions of the tangency portfolio, given a risk-free rate R_f, are obtained as

$$\mathbf{w}^{\text{TGP}} = \frac{\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - R_f)}{\boldsymbol{\iota}'\boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu} - R_f)}.$$

A central consequence of the efficiency of all portfolios along the CML is that it is optimal for all investors (who share a common view about $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$) to



◀ **Figure 2.2.** *Efficient frontier obtained from four assets specified in the text; the Global Minimum Variance (GMV) portfolio has a lower risk (as measured by the standard deviation of returns) than any individual asset, showing the benefits of diversification.*

hold the *tangency portfolio in some proportion*. Investors only differ in their exposure to it, or alternatively, in how they allocate their holdings between the risk-free and tangency portfolio. This result was originally established by Tobin (1958) (see also Merton (1990, ch. 2)) and is an example of *separation* or *mutual fund* theorems.*

In the presence of a risk-free asset, portfolio optimization problems can be formulated without insisting on the "sum-to-one" constraint (2.4), since the unallocated fraction of capital, $1 - \mathbf{w}'\boldsymbol{\iota}$, can be invested in the risk-free asset (or assumed to be borrowable at the risk-free rate in the case of a negative fraction).

Geometrically, from Fig. 2.2, the tangency portfolio can also be seen to maximize the *Sharpe ratio* (Sharpe 1966, 1994), defined as the expected portfolio excess return (over the risk-free rate R_f) per unit of portfolio return standard deviation,

$$\mathrm{SR} \triangleq \frac{\mu_P - R_f}{\sigma_P},$$

with μ_P and σ_P given by eq. (2.1). A formal derivation of the relationship between the Sharpe ratio and the tangency portfolio appears in §A.2/p. 296.

*This result also serves as a foundation for the celebrated Capital Asset Pricing Model (CAPM), which assumes, among other things, that all investors do share common views about $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and examines equilibrium consequences; see §2.1.7/p. 25.

2.1.4 Utility Maximization

Problem (2.2) does not specify what the "appropriate" level of target return ρ should be; this question should be decided by the investor and is a direct function of the risk s/he is *willing* and *able* to bear. Markowitz (1959) introduces a formulation wherein the investor's expected utility is directly maximized. He considered the following quadratic form, written in terms of the portfolio return R_P,

$$U_\lambda(R_P) = R_P - \frac{\lambda}{2} R_P^2,$$

where λ is represents the investor's *risk aversion*, and in this context quantifies how the investor is willing to trade each incremental unit of expected return against a corresponding increase in variance of return.*

A rational decision maker would seek to maximize its *expected utility*, which is computed as

$$\begin{aligned} \mathbb{E}\left[U_\lambda(R_P)\right] &= \mu_P - \frac{\lambda}{2}\sigma_P^2 \\ &= \mathbf{w}'\boldsymbol{\mu} - \frac{\lambda}{2}\mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w}, \end{aligned}$$

where $\mathbf{w}$ is, as above, the weight given on each asset within the portfolio and μ_P and σ_P^2 are respectively the mean and variance of the portfolio return distribution, given by eq. (2.1). The expected quadratic utility maximization problem is then written as

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \mathbf{w}'\boldsymbol{\mu} - \frac{\lambda}{2}\mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w} \tag{2.7}$$

$$\text{subject to} \quad \mathbf{w}'\boldsymbol{\iota} = 1. \tag{2.8}$$

When no further constraint is imposed, an analytical solution for $\mathbf{w}^*$ is easily found by introducing Lagrange multipliers, similarly to the solution for problem (2.2).†

Proposition 3 *The unconstrained minimum-variance portfolio* (2.2)–(2.3) *and maximum quadratic utility* (2.7) *formulations are equivalent.*

Proof The equality constraint (2.3) is incorporated in the minimum-variance objective (2.2) through an unconstrained Lagrange multiplier $\nu \in \mathbb{R}$, yielding the problem

$$\min_{\mathbf{w}} \frac{1}{2}\mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w} - \nu(\mathbf{w}'\boldsymbol{\mu} - \rho),$$

*Many formulations of utility theory focus on the utility of *terminal wealth*, instead of the portfolio return; Markowitz explicitly considers the latter (e.g. Markowitz 1959, p. 208), and this convention is almost universally followed in mean-variance problems. An alternative formulation of quadratic utility in terms of terminal wealth would slightly change the resulting equations.

†See, e.g. Chapados (2000) for a derivation.

with first-order conditions for optimality given by $\boldsymbol{\Sigma}\,\mathbf{w} - \nu\boldsymbol{\mu} = 0$, yielding optimal solution

$$\mathbf{w}^* = \nu\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \tag{2.9}$$

where ν is found by substitution as $\nu = \frac{\rho}{\boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}$.

Consider, on the other hand, the first-order optimality conditions of problem (2.7), $\boldsymbol{\mu} - \lambda\boldsymbol{\Sigma}\,\mathbf{w} = 0$, yielding optimal solution

$$\mathbf{w}^* = \frac{1}{\lambda}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}. \tag{2.10}$$

Comparing eq. (2.9) and (2.10), it suffices to take $\lambda = \boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}/\rho$ to obtain the equivalence. ■

This result confirms that in order to target a higher expected portfolio return ρ, the investor must exhibit a lower risk aversion.

Obviously, quadratic utility is but one of a number of utility functions that have been proposed to model the behavior of economic agents. The more general problem is easily written in terms of expected utility maximization,

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \int_{\mathbf{R}} U(\mathbf{w}'\mathbf{R})\,\mathrm{d}P(\mathbf{R}), \tag{2.11}$$

subject to the budget constraint (2.8), where $U(\cdot)$ is a utility function and $P(\mathbf{R})$ is the next-period return distribution. In particular, Mossin (1968) proves that constant relative risk aversion (CRRA) functions* are the only ones permitted if constant asset proportions are to be optimal, i.e. the investment in the risky asset does not depend on the level of initial wealth. Merton (1969) establishes the same result in a continuous-time setting. Moreover, Campbell and Viceira (2002) strongly argue in favor of CRRA utilities on the basis of the long-run observed behavior of the economy. However, for a large number of utility functions and "reasonable" return distributions, several studies (Levy and Markowitz 1979; Kallberg and Ziemba 1983) have established that single-period optimal portfolios under quadratic utility are very close to those obtained under alternative utilities.

A special case of some interest is the logarithmic utility, defined as $U(R) = \log(1 + R)$. This utility function is maximized by considering a Taylor series expansion of $1 + R$ around $R = 0$,

$$\log(1 + R) = R - \frac{R^2}{2} + O(R^3).$$

*For a utility function $U(W)$, the Arrow–Pratt measure of relative risk aversion (Arrow 1965; Pratt 1964) is defined as

$$\mathrm{RRA}(W) = -\frac{WU''(W)}{U'(W)}.$$

A CRRA utility function is one for which $\mathrm{RRA}(W)$ is a constant independent of W. Such functions are sometimes said to exhibit *iso-elastic marginal utility*.

For relatively small returns, this is seen to be equivalent to the maximization of quadratic utility, problem (2.7), with $\lambda = 1$. The optimal weights under this utility function are given precisely by the tangency portfolio for a risk-free rate of zero (which also maximizes the Sharpe Ratio, see §A.2/p. 296). This property led some authors to confer a special aura to the logarithmic utility as being somehow "better", a point discussed, and found to be fallacious in a multiperiod setting, by Merton and Samuelson (1974). We return to the logarithmic utility in §2.2.4/p. 49.

Some utility functions have been proposed to incorporate parameter estimation uncertainty, the subject of *robust optimization*, which is covered in §2.1.8/p. 36.

2.1.5 Risk Measures

The exposition so far assumes that the investor considers the variance of the portfolio return distribution to be an adequate measure of risk. This measure has the major shortcoming that it considers positive return surprises to be as equally unpleasant as negative return surprises, a property that would surely be dismissed by most real-world investors! A number of alternative measures have been proposed throughout the years that attempt to quantify *portfolio downside risk*, starting with Markowitz's original treatment of the semivariance. This section briefly reviews the most significant possibilities. Nawrocki (1999) surveys the field more extensively.

Semivariance

Semivariance was originally considered by Markowitz (1959, Chapter 9) as a simple measure of downside risk. Whereas the variance is a symmetrical measure, semivariance only considers movements that fall below the mean; as such, its value depends on the *skewness* (third moment) of the distribution. For a scalar random variable X with mean μ, semivariance is defined as

$$\sigma^2_{\text{min}} = \mathbb{E}\left[\min\left[X - \mu, 0\right]^2\right].$$

This measure can be used instead of portfolio variance in Problem (2.2). Although there is no closed-form solution to the mean-semivariance problem, Jin, Markowitz, and Zhou (2006) establish the existence of the one-period mean-semivariance efficient frontier and review the literature examining its applications. Furthermore, Estrada (2007) provides an approximation to the semivariance that lends itself well to analytical solutions and reports good results on a number of problems.

Roy's Safety First

The Roy (1952) "safety-first" criterion puts portfolio risk in a more concrete setting than Markowitz' consideration of the second moment of returns.

As Roy argued, the investor first decides on a minimum acceptable return that would ensure the preservation of a desired portion of his capital; he then proceeds with portfolio optimization by minimizing the probability of experiencing a return below the "disaster level". Let R_0 be the investor's minimum acceptable return and consider the problem

$$\begin{aligned} \text{minimize} \quad & P(R_P \leq R_0) \\ \text{subject to} \quad & \mathbf{w}'\boldsymbol{\iota} = 1 \qquad \text{(budget)}. \end{aligned}$$

Since the return distribution probability is not known precisely, this minimization may appear unfeasible. However, by Chebyshev's inequality, we have

$$P(R_P \leq R_0) \leq \frac{\sigma_P^2}{(\mu_P - R_0)^2},$$

which, taking square roots, yields the approximate problem

$$\min_{\mathbf{w}} \frac{\sigma_P}{\mu_P - R_0}$$

subject to the budget constraint. If the R_0 is the risk-free rate, this problem is equivalent to maximizing the Sharpe ratio (Sharpe 1966).

Value-at-Risk

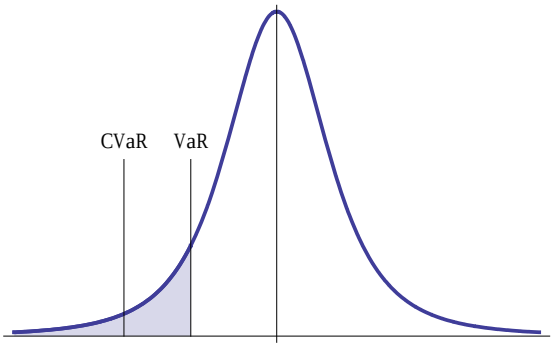
Value-at-Risk (VaR) was developed by JP Morgan in the early 1990's and made popular in a widely-circulated technical document (RiskMetrics 1996) and associated software product. Intuitively, the level-α VaR (e.g. $\alpha = 95\%$) of a portfolio over a certain time horizon h is the portfolio return R_P such that the fraction α of returns will be better than R_P over the horizon. More formally, the level-α VaR of a portfolio is defined as the $1-\alpha$-percentile of the portfolio return distribution,

$$\text{VaR}_\alpha(R_P) = \inf_R \{R : P(R_P \geq R) \geq \alpha\},$$

where all returns are computed over horizon h. The location of the VaR of an hypothetical asset return distribution, and its relationship to the CVaR (treated next) is shown in Fig. 2.3.

Value-at-Risk is regarded as a more plausible measure of portfolio risk than the variance since it accounts (in theory) for skewness and kurtosis in the return distribution.* In addition to its origins in risk management, it has received wide attention in a portfolio choice context where the VaR simply substitutes for the variance as the risk measure (Alexander and Baptista 2002; Mittnik, Rachev, and Schwartz 2003; Chow and Kritzman 2002; Chapados 2000).

*In practice, it is common to compute the VaR under a normal approximation due to its analytical tractability, which of course disregards higher-order moments in the underlying true distribution.



◀ **Figure 2.3.** *90% Value-at-Risk (VaR) and Conditional Value-at-Risk (CVaR) for a Student $t(3)$ distribution. For fat-tailed distributions, the CVaR point can represent an expected loss much more significant than the VaR.*

Conditional Value-at-Risk

In spite of its wide use, the VaR, as a measure of risk, suffers from a major defect: its lack of *subadditivity* (Artzner, Delbaen, Eber, and Heath 1999). For a risk measure ρ applied to portfolios P_1 and P_2, subadditivity is satisfied if

$$\rho(P_1 + P_2) \leq \rho(P_1) + \rho(P_2),$$

which is a statement of the benefits of diversification—the risk of a diversified portfolio cannot be more than the risk of any of its constituents. That the VaR does not satisfy this property can lead to a number of counterintuitive results, particularly for firm-wide risk management, where it can appear that a more diversified portfolio exhibits a higher risk (Rau-Bredow 2004).

A closely related measure that does satisfy subadditivity is the *conditional value at risk* (CVaR)—also called *expected shortfall* or *expected tail loss*—defined as the expected return conditional on observing a return lower than the VaR:

$$\mathrm{CVaR}_\alpha(R_P) = \mathbb{E}[R_P \,|\, R_P < \mathrm{VaR}_\alpha(R_P)],$$

where, as for the VaR, the returns are computed over a given time horizon h. In Fig. 2.3, this corresponds to an expectation taken within the shaded area. In a portfolio context, the CVaR has been studied by Krokhmal, Palmquist, and Uryasev (2002) and Consigli (2004).

Other Measures

In the past few years, there has been an explosion of alternative risk measures based on the modeling of tail phenomena (e.g. Malevergne and Sornette 2005a). Although it is not our focus to describe them in depth, Rachev, Menn, and Fabozzi (2005) provide a good survey of the relevant literature, especially of measures related to portfolio selection. Farinelli, Rossello, and Tobiletti (2006) provide computational portfolio allocation results comparing eleven alternative performance measure ratios.

2.1.6 Additional Constraints

Portfolio optimization problems, regardless of the form of the objective function or type of risk measure, are often solved with a number of constraints that attempt to capture *a priori* knowledge that the analyst possesses on what should be "good" solutions, embody investment objectives of the fund, or comply with regulatory requirements. It should be noted that with most of these constraints, Problem (2.2) can no longer be solved analytically but must instead be tackled with quadratic programming (Luenberger and Ye 2007; Bertsekas 2000) or mixed-integer programming (Wolsey and Nemhauser 1999). Constraints also play a *regularization* role that can serve to mitigate sampling variance and estimation error in the mean return and risk forecasts; this is covered in §2.1.8/p. 31.

Some of the more common constraints are as follows. More comprehensive treatments appear in Fabozzi, Focardi, and Kolm (2006) and Qian, Hua, and Sorensen (2007). In line with the first reference, the rest of this section makes use of the following notation: we denote the current holdings of an investor by $\mathbf{w}_0$, the target holdings to be invested over the next period (i.e. the variables resulting from optimization) by $\mathbf{w}$, and their difference (the traded amount in each asset) by $\mathbf{x} = \mathbf{w} - \mathbf{w}_0$.* Furthermore, let $\mathbf{p}_0$ be the current price vector of the assets, and W_0 the current total portfolio value. The amount to be invested in asset i is given by $W_0\mathbf{w}_i$ and the number of shares† held is $n_i = W_0\mathbf{w}_i/\mathbf{p}_{0i}$.

No Short-Sales Constraint This corresponds to the requirement that all portfolio weights be non-negative, namely

$$\mathbf{w}_i \geq 0, \quad \text{for all } i,$$

thereby prohibiting selling assets short. Regulatory constraints placed on mutual fund managers often mandate such a constraint. Markowitz' original formulation of the portfolio choice problem included those constraints as an integral part of his solution method, and many introductory treatments of the theory‡ include them by default, despite the impossibility of deriving an analytical solution for the optimal portfolio weights in their presence.§

*The absolute traded amount, $|\mathbf{x}| = |\mathbf{w} - \mathbf{w}_0|$, shall be of significance, especially when considering transaction costs. The usual way of incorporating a term of this kind in a mathematical program is to introduce two variables,

$$\mathbf{x}^+ = \mathbf{w} - \mathbf{w}_0 \qquad \text{and} \qquad \mathbf{x}^- = \mathbf{w}_0 - \mathbf{w}$$

along with the constraints

$$\mathbf{x}^+ \geq \mathbf{0} \qquad \text{and} \qquad \mathbf{x}^- \geq \mathbf{0}$$

and use the sum $\mathbf{x}^+ + \mathbf{x}^-$ whenever $|\mathbf{x}|$ appears.

†Assuming stocks as the assets.

‡E.g. Bodie, Kane, and Marcus (2004).

§Non-negativity constraints can be seen as the "great divide" in optimization between analytical and non-analytical solutions; in the case of portfolio optimization, the latter

Turnover and Transaction Costs Constraints For large institutional portfolios, transaction costs can represent a sizable portion of total operational costs, especially for funds that take an *active management* (Grinold and Kahn 2000) approach as opposed to a passive index-tracking objective. As such, we may incorporate constraints that attempt to minimize the relative or dollar turnover on individual assets, respectively

$$|\mathbf{x}_i| \leq U_i \qquad \text{and} \qquad W_0|\mathbf{x}_i| \leq \tilde{U}_i,$$

or the complete portfolio

$$\sum_i |\mathbf{x}_i| \leq U_P.$$

It is also possible to directly incorporate *transaction costs* into the objective function as a term to be minimized. In its simplest form, a transaction cost model simply imposes a proportional cost on the absolute value of traded quantities,

$$\text{prop cost}_i = W_0 \chi_i |\mathbf{x}_i|,$$

and the total portfolio cost given by

$$\text{prop cost}_P = W_0 \sum_i \chi_i |\mathbf{x}_i|, \tag{2.12}$$

where χ_i is the proportional cost of trading asset i. Assuming all χ_i and W_0 are nonnegative, prop cost_P is nonnegative and hence the imposition of transaction costs penalizes portfolio performance. To understand their consequence on realized returns, let $\mathbf{p}_1$ be the asset prices at the end of the investment period and consider the relative return on asset i,

$$r_i = \frac{\mathbf{p}_{1i} - \mathbf{p}_{0i}}{\mathbf{p}_{0i}}.$$

Transaction costs affect portfolio return as

$$\begin{aligned} \tilde{r}_i &= \frac{\mathbf{p}_{1i} - \mathbf{p}_{0i} - W_0 \chi_i |\mathbf{x}_i|}{\mathbf{p}_{0i}} \\ &= r_i - \frac{W_0}{\mathbf{p}_{0i}} \chi_i |\mathbf{x}_i| \quad = \quad r_i - n_i \chi_i |\mathbf{x}_i| \quad = \quad r_i - \tilde{\chi}_i |\mathbf{x}_i|, \end{aligned}$$

where it is obvious that they adjust the portfolio relative return by a term proportional to the traded amount. Their effect can then directly be incorporated into the objective function for the quadratic utility maximization formulation, yielding the problem

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \mathbf{w}'\boldsymbol{\mu} - \tilde{\chi}'|\mathbf{w} - \mathbf{w}_0| - \lambda \mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w} \tag{2.13}$$

$$\text{subject to} \quad \mathbf{w}'\boldsymbol{\iota} = 1. \tag{2.14}$$

require, as mentioned above, solution by quadratic programming.

The proportional costs structure is, however, only a starting point. As pointed out by Kissell and Glantz (2003), the totality of trading costs can be broken down according to an elaborate taxonomy that includes *explicit* (measurable) costs as well as more insidious *implicit* ones. Without delving into an intricate description, we can summarize them as follows:

Explicit Costs They include *fixed costs*, in the form of commissions (as outlined above) and fees (custodial fees, transfer fees). They also include *variable costs*, in the form of bid–ask spread (the difference between the price at which one can buy versus sell) and taxes.*

Implicit Costs They include *delay cost* (time between which a decision is made—for instance, by an allocation committee—and the actual trade is brought to the market), *price movement risk* (effect of underlying trends affecting the asset to be traded), *market impact costs* (deviation of the transaction price from the market price that would have prevailed had the trade not occurred), *timing risk* (cost attributable to general market volatility), *opportunity cost* (cost of not trading or not completing a trade).

Some of the implicit costs may not be costs at all but the source of trading profits depending on market conditions. A study by Wagner and Edwards (1998) shows that the price impact of a liquidity-demanding trade† averages −103 basis points‡ on a set of some 700,000 trades by more than 50 management firms in 1996, whereas the price impact of a liquidity-supplying trade generated *profits* of +36 basis points. In a liquidity-neutral market, the average price impact was −23 basis points. The effects of other implicit costs can likewise be decomposed according to market conditions.

There is a vast literature on transaction costs models, including how realistic non-linear models of costs can be incorporated in asset-allocation models. This literature is well reviewed by Fabozzi, Focardi, and Kolm (2006, ch. 3).

Maximum Holdings Constraint To ensure that the portfolio is not overly concentrated in a single asset, we can impose a constraint of the form

$$\mathbf{L} \leq \mathbf{w} \leq \mathbf{U},$$

where $\mathbf{L}$ and $\mathbf{U}$ are vectors specifying, respectively, the allowable lower and upper bounds for each asset. Likewise, we can ensure a sector $\mathcal{S} =$

*The proportional costs structure introduced previously can be seen as an adequate model of bid–ask spread, the most significant explicit cost for an institutional investor.

†For example, a "buy" trade executed when there are significantly more buyers than sellers.

‡A *basis point* (bp) is one hundredth of one percent, i.e. 100 bp = 1%.

$\{i_1, i_2, \ldots, i_n\}$ (a set of asset indices) is not unduly weighted in the portfolio by imposing

$$L_{\mathcal{S}} \leq \sum_{i \in \mathcal{S}} \mathbf{w}_i \leq U_{\mathcal{S}},$$

with $L_{\mathcal{S}}$ and $U_{\mathcal{S}}$ denoting, respectively, the minimum and maximum exposure to the sector.

Maximum Tracking Error and Factor Exposure Constraint The performance of portfolio managers is often compared to that of a *benchmark* such as the S&P 500 (Grinold and Kahn 2000). Depending on the fund's style, the manager may seek to replicate the benchmark as closely as possible (using, for instance, a smaller number of assets than the benchmark), or to provide additional performance (the so-called "alpha") at the expense of taking on *active risk*, namely, deviating from the benchmark. This risk is quantified by the *tracking error*, defined next. Assume that the benchmark's and fund's investable universe are the same and that the (random) asset returns are given by $\mathbf{R}$. Let $\mathbf{w}_b$ denote the benchmark weights, $\mathbf{w}$ the decision variables, and R_B and R_P denote, respectively, the benchmark and portfolio returns,

$$R_B = \mathbf{w}_B' \mathbf{R} \qquad \text{and} \qquad R_P = \mathbf{w}' \mathbf{R}.$$

The tracking error is simply the variance of the return difference between the benchmark and the invested portfolio,*

$$\begin{aligned} \mathrm{TE}_P &= \mathrm{Var}[R_P - R_B] \\ &= \mathrm{Var}[\mathbf{w}_B' \mathbf{R} - \mathbf{w}' \mathbf{R}] \\ &= (\mathbf{w}_B - \mathbf{w})' \boldsymbol{\Sigma} (\mathbf{w}_B - \mathbf{w}), \end{aligned}$$

with $\boldsymbol{\Sigma}$ the asset return covariance matrix. A quadratic tracking error constraint of the form

$$(\mathbf{w}_B - \mathbf{w})' \boldsymbol{\Sigma} (\mathbf{w}_B - \mathbf{w}) \leq \sigma_{\mathrm{TE}}^2$$

can then be imposed to limit active risk. Note that this does not limit *total risk*, which would require additional constraints (Jorion 2003).

In an analogous manner, one can restrict exposure to specific risk factors. Suppose that we posit the following decomposition for *explaining* the return of asset i as a linear combination of factors (additional background on factor models is given in §2.1.7/p. 24),

$$R_i = \alpha_i + \sum_{j=1}^{M} \beta_{i,j} F_j + \varepsilon_i,$$

*More accurately, *tracking error* is usually reserved for the square-root of this variance, but for notational simplicity, we shall omit the square-roots in this overview.

where F_j is the random "return" associated with factor j* during the period, and $\beta_{i,j}$ is the exposure of asset i to factor j. This is written more succinctly as

$$\mathbf{R} = \boldsymbol{\alpha} + \mathbf{BF} + \boldsymbol{\varepsilon}$$

with $\mathbf{B}$ and $\mathbf{F}$ respectively the matrix of factor exposures and the vector of one-period factor returns. This yields a portfolio return, given asset weights $\mathbf{w}$, of

$$R_P = \mathbf{w}'\boldsymbol{\alpha} + \mathbf{w}'\mathbf{BF} + \mathbf{w}'\boldsymbol{\varepsilon}.$$

The exposure of the portfolio to factor j is given by $\sum_i \mathbf{w}_i\beta_{i,j}$. Bound or equality constraints may be placed on this exposure; for example, to ensure an *ex ante* neutral exposure to factor j one may impose

$$\sum_i \mathbf{w}_i\beta_{i,j} = 0.$$

Such constraints are commonly used in so-called "long-short equity" hedge funds, which are designed to be neutral to overall market fluctuations.†

Transaction Size, Cardinality and Round Lot Constraints The following class of constraints is of a combinatorial nature and necessitates solution by *mixed integer programming* methods (Wolsey and Nemhauser 1999). For convenience, we define the vector $\boldsymbol{\delta}$ of binary indicator variables

$$\delta_i = \begin{cases} 1, & \text{if } \mathbf{w}_i \neq 0, \\ 0, & \text{if } \mathbf{w}_i = 0, \end{cases} \qquad i = 1, \ldots, N,$$

where each element specifies whether a position is being taken in the corresponding asset.

A first class of combinatorial constraints aims at eliminating positions that are too small; such positions are often the result of a traditional unconstrained mean–variance optimization. The manager can require

$$|\mathbf{w}_i| \geq \boldsymbol{\delta}_i \mathbf{L}_{\mathbf{w}_i},$$

where $\mathbf{L}_{\mathbf{w}_i}$ is the minimum (relative) position size allowed for asset i. Likewise a limit can be set on portfolio trades

$$|\mathbf{x}_i| \geq \boldsymbol{\delta}_i \mathbf{L}_{\mathbf{x}_i},$$

with $\mathbf{L}_{\mathbf{x}_i}$ the minimum allowed trade size for asset i.

*For stocks, examples of likely factors would be the return on a broad market index, the return difference between growth and value stocks, and the return difference between large- and small-capitalization stocks; see §2.1.7/p. 24.

†For a factor-neutral constraint to make sense, the exposures $\beta_{i,j}$ must be standardized to have a mean of zero across assets.

Next, *cardinality* constraints can be useful in problems that seek to replicate a benchmark using a smaller number of assets than the original universe. This may take the form of

$$\boldsymbol{\delta}'\boldsymbol{\iota} \leq K$$

where K is the maximum number of allowable assets. The impact of cardinality constraints on the shape of the efficient frontier is studied by Chang, Meade, Beasley, and Sharaiha (2000).

Finally, *round lot* constraints account for the fact that market-traded instruments are not infinitely divisible (contrarily to idealizations of finance theory)—it is common for stocks to be traded in multiples of 100 shares or more. If the lot size for asset i is given by the constant κ_i and the desired number of lots by η_i (an integer decision variable), we can enforce

$$W_0\mathbf{w}_i = \kappa_i\eta_i\mathbf{p}_{0i}, \qquad \eta_i \in \mathbb{Z}.$$

In general, when imposing round lots, the budget constraint, $\sum_i \mathbf{w}_i = 1$, may no longer be satisfiable; in this case, one may settle for an approximate budget constraint, expressed as

$$\begin{aligned} \frac{1}{W_0}\sum_i \kappa_i\eta_i\mathbf{p}_{0i} + \xi^+ - \xi^- &= 1, \\ \xi^+,\, \xi^- &\geq 0, \\ \eta_i &\in \mathbb{Z}, \end{aligned}$$

where ξ^+ and ξ^- are "slack variables" to be minimized (by incorporating them in the objective function). Formulations of this type are analyzed by Kellerer, Mansini, and Speranza (2000).

2.1.7 Forecasting Models

Markovitz's method of portfolio construction is silent on how the required expected next-period asset returns and covariances are to be obtained. This section reviews the most commonly-used approaches in practice, starting with *factor models* and their uses in covariance modeling and expected return forecasts. We then briefly cover other expected-return forecasting approaches for equities, mostly based on *dividend discount models* and accounting ratios. Finally, extensive experience with mean-variance criteria suggest that they are extremely sensitive to parameter estimation error—very small changes in the forecasts can yield enormous changes in "optimal" portfolio weights, leading to doubt about the validity of the portfolios and possible considerable rebalancing costs when the decisions are implemented. This naturally paves the way for robust estimation methods and Bayesian approaches; we cover some of the methods that have been suggested to counter portfolio instability.

Factor Models

Factor models seek to explain the *cross-section*[*] of asset returns by a simple affine relationship, where the return of asset i over the period is decomposed into the return of more elemental *factor returns* F_j,

$$R_i = \alpha_i + \sum_{j=1}^{M} \beta_{i,j} F_j + \varepsilon_i, \tag{2.15}$$

where α_i is a regression constant, $\beta_{i,j}$ are the factor exposures, and ε_i is a zero-mean random unexplained component uncorrelated with factor returns.[†]

The grandfather of factor models is the Capital Asset Pricing Model (CAPM) of Sharpe (1964), Lintner (1965) and Mossin (1966); this model is generally derived from equilibrium considerations as a *positive theory* of collective investor behavior,[‡] but we shall merely regard it as a simple one-factor model. It expresses the expected excess return[§] on asset i as a linear function of the return of the overall market portfolio, R_M,

$$\mathbb{E}[R_i - R_f] = \beta_i \mathbb{E}[R_M - R_f],$$

where, under the CAPM assumptions, α_i is identically zero.[¶]

It has long been understood, at least since Merton (1973), that there exists the possibility that additional sources of *priced risk*, on top of the market portfolio, could impact expected asset returns. Generalizations of the CAPM are obtained in the context of the Arbitrage Pricing Theory (APT) of Ross (1976).[‖] Assume that asset returns are distributed according to the factor structure of eq. (2.15), along with

$$\begin{aligned} \mathbb{E}[\varepsilon_i] = \mathbb{E}[F_k] &= 0 \\ \mathbb{E}[\varepsilon_i \varepsilon_j] = \mathbb{E}[\varepsilon_i F_j] = \mathbb{E}[F_i F_j] &= 0, \quad i \neq j \\ \mathbb{E}[\varepsilon_i^2] &= \sigma^2 < \infty. \end{aligned}$$

[*] As opposed to the time-series characteristics.

[†] It should be noted that what this literature refers to as *factors* almost exclusively consist of observable variables, what would simply be called explanatory or input variables in a more traditional statistical context. Latent factors are always referred to as such.

[‡] In other words, it seeks to establish what consequences would arise if every investor behaved according to a set of hypotheses that include Markowitz's rules for portfolio choice among others.

[§] The return earned over the risk-free rate.

[¶] Starting from the late-1960's, a huge literature has emerged aiming at testing the validity of the CAPM; see Campbell, Lo, and MacKinlay (1997) for an overview.

[‖] Technically, the CAPM is derived from equilibrium considerations whereas the APT is derived from a more fundamental "absence of arbitrage" principle; these minutiae make little difference from a statistical estimation standpoint.

In this context, in the absence of arbitrage and under some technical conditions, Ross showed that the excess return on asset i is given by

$$\mathbb{E}[R_i - R_f] = \sum_{j=1}^{K} \beta_{i,j} \mathbb{E}[F_j - R_f].$$

Under the APT, each factor represents a systematic priced risk (a risk for which investors are seeking compensation), and the factor exposures $\beta_{i,j}$ quantify the *market price* of those risks (how much the investor is compensated in expected return for taking on a unit of risk).

Ross remains silent on how factors should be chosen. In addition to the CAPM market portfolio factor, several *pricing anomalies* have been documented in the 1980's and early 1990's suggesting additional factors, including long-run price reversal (De Bondt and Thaler 1985), short-run price momentum (Jegadeesh and Titman 1993), and a variety of effects due to firm size (market equity, ME, the stock price times the number of shares), earnings to price ratio (E/P), cash-flow to price ratio (C/P), book value to market value (BE/ME), and past sales growth (Banz 1981; Basu 1983; Rosenberg, Reid, and Lanstein 1985; Lakonishok, Shleifer, and Vishny 1994). These results built up to an influential series of papers by Fama and French (1992, 1993, 1995, 1996), who show that the following two additional factors summarize well a number of empirical findings:

High-Minus-Low (HML) The difference between the return on a portfolio of high-book-to-market stocks and the return on a portfolio of low-book-to-market stocks.*

Small-Minus-Big (SMB) The difference between the return on a portfolio of small stocks and the return on a portfolio of large stocks.

Put together, Fama and French argue that a model of the form

$$\mathbb{E}[R_i - R_f] = \beta_i \mathbb{E}[R_M - R_f] + s_i \mathbb{E}[\text{SMB}] + h_i \mathbb{E}[\text{HML}]$$

can account for a large fraction of the cross-section of returns, and obtain times-series regression R^2 in the 0.90–0.95 range. The only factor significantly unaccounted for is the short-run price momentum, which is empirically analyzed by Carhart (1997).

Since the late 1990's, several large commercial factor models have become available, the best known of which is perhaps Barra's fundamental multi-factor risk model for United States equities (Barra 1998), which includes 13 risk indices and 55 industry groups.

*The precise definition is slightly technical and appears in Fama and French (1996).

Factor Models in Covariance Matrix Estimation

The estimation of covariance matrices for portfolios of many assets is a hard problem. As an illustration, consider the Russell 1000 index, whose sample covariance matrix

$$\hat{\mathbf{\Sigma}} = \frac{1}{T-1}\sum_{t=1}^{T}(\mathbf{R}_t - \hat{\boldsymbol{\mu}})(\mathbf{R}_t - \hat{\boldsymbol{\mu}})'$$

contains 500,500 distinct entries;* an analysis with the tools of *random matrix theory* shows that for such large matrices, only a few eigenvalues of the sample covariance matrix carry information, the rest being the result of noise (Laloux, Cizeau, Bouchaud, and Potters 1999; Malevergne and Sornette 2005b). This observation gave rise to a number of schemes to add structure to the estimator, often relying on *shrinkage methods* that attempt to find an optimal compromise between a restricted and unrestricted estimators (§2.1.8/p. 31).

An obvious application of factor models is to the estimation of covariance matrices. This approach can be traced back to a suggestion by Sharpe (1963), and relies on the factor decomposition of eq. (2.15). Assume that firm-specific residual returns, ε_i, are uncorrelated for two different firms,

$$\mathbb{E}[\varepsilon_i \varepsilon_j] = \begin{cases} 0, & i \neq j, \\ \sigma_i^2, & i = j. \end{cases}$$

The covariance between returns R_i and R_j is obtained from eq. (2.15) as

$$\begin{aligned} \operatorname{Cov}[R_i, R_j] &= \sum_{k=1}^{M} \operatorname{Cov}[\beta_{i,k}F_i, \beta_{j,k}F_j] + \operatorname{Cov}[\varepsilon_i, \varepsilon_j] \\ &= \sum_{k=1}^{M} \beta_{i,k}\beta_{j,k} \operatorname{Cov}[F_i, F_j] + \delta_{i,j}\sigma_i^2, \end{aligned}$$

where $\delta_{i,j}$ is the Kronecker delta. This expression illustrates that under a factor model of returns, the covariance between arbitrary assets depends only on the *covariance matrix between the individual factors*, which (for the small number of factors used in practice) is a much more tractable quantity to estimate with statistical reliability. Current methods for covariance modeling are reviewed by Fabozzi, Focardi, and Kolm (2006) and Qian, Hua, and Sorensen (2007).

Factor Models in Expected Return Estimation

Forecasting expected asset returns is recognized as notoriously difficult — so much so that this apparent unforecastability gave rise to the Efficient

*Obtained as $1000 \times 1001/2$.

Market Hypothesis (EMH) and a famous proof that prices should fluctuate randomly (Cootner 1964; Samuelson 1965; Fama 1970). Empirically, it is often observed that the simplest predictors, a constant based on the historical average return or even the constant *zero*,* perform the best out of sample. More recently, with advances in computing power and improvements in the quality and quantity of available data, mounting evidence has started to accumulate in favor of some *very small* forecastability (Lo and MacKinlay 1999), possibly arising from market imperfections. However, exploiting any residual forecastability, especially when accounting for trading costs, remains of the utmost challenge.

Factor models can provide some direction in this respect and are generally used by relating the returns at time t with the observed factors at the same time, and then positing a dynamical model for making forecasts of the factors themselves. It is common to utilize a Vector Autoregressive (VAR) model for establishing the dynamics (Hamilton 1994), yielding an overall forecasting model specified as

$$\begin{aligned}\mathbf{R}_t &= \boldsymbol{\alpha} + \boldsymbol{\beta}'\mathbf{F}_t + \boldsymbol{\varepsilon}_{\mathbf{R},t}\\ \mathbf{F}_{t+1} &= \mathbf{a} + \mathbf{B}\mathbf{F}_t + \boldsymbol{\varepsilon}_{\mathbf{F},t+1},\end{aligned}$$

where $\mathbf{a}$ is a vector and $\mathbf{B}$ is a matrix of first-order autoregression factors.

An example that has received wide attention is the forecastability of stock returns by the dividend yield.† Brandt (2004) estimates the following parameters for the quarterly returns of the value-weighted CRSP‡ index

$$\begin{bmatrix} r^e_{t+1} \\ d_{t+1} - p_{t+1} \end{bmatrix} = \begin{bmatrix} 0.2049 \\ (0.0839) \\ -0.1694 \\ (0.0845) \end{bmatrix} + \begin{bmatrix} 0.0568 \\ (0.0249) \\ 0.9514 \\ (0.0251) \end{bmatrix} (d_t - p_t) + \begin{bmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{bmatrix} \tag{2.16}$$

$$\begin{bmatrix} \varepsilon_{1,t+1} \\ \varepsilon_{2,t+1} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.0062 & -0.0060 \\ -0.0060 & 0.0063 \end{bmatrix} \right),$$

where r^e_t denotes the log excess return of the index and $d_t - p_t$ is the log dividend yield, computed from the log of the trailing-twelve-month sum of monthly dividends d_t and the current index level p_t.§ In parenthesis are the

*Which is surprisingly effective in the case of daily stock returns.

†The first evidence is presented in Campbell and Shiller (1988) and Fama and French (1988); Campbell (1991) presents an interesting decomposition of stock returns wherein he shows that unexpected stock returns must be associated with changes in expected future dividends or expected future returns, and attributes a third of the variance in U.S. unexpected returns over the 1927–88 period to the first component, a third to the second, and the final third to their covariance. For use of the dividend yield in an asset allocation context, see e.g. Kandel and Stambaugh (1996) and Brennan, Schwartz, and Lagnado (1997).

‡Center for Research in Security Prices, based at the University of Chicago; `www.crsp.com`.

§The estimation period in this example is from April 1952 to December 1996, and the results are fairly stable across different estimation periods.

Newey and West (1987) standard errors. These results serve to illustrate that whatever forecastability remains, although statistically significant over a long sample, remains low.

Other Expected Return Forecasting Models

A different angle on forecasting models for equities is provided by the *fundamental analysis* of a firm's fair value. The starting point in this line of study is the *dividend discount model* (DDM), introduced by Williams (1938), stating that the price of one share of stock should be given by the sum of discounted future dividend payments,

$$P_t = \mathbb{E}_t \left[\sum_{\tau=1}^{\infty} \frac{D_{t+\tau}}{(1+R_{t+\tau})^{\tau}} \right], \tag{2.17}$$

where D_t is the dividend to be paid in (future) period t and R_t are discount rates.* It should be noted that the discount rate is generally higher than the prevailing risk-free rate and reflects the market's expectations on the prospects of future dividend payments; a greater risk on the dividend stream entails a higher discount rate. In other words, it can be viewed as the rate of return that investors *require* for bearing the risk of holding the equity. Consider a simplification wherein we keep the discount factor constant at R and assume a constant growth rate g for dividends,† $D_{t+1} = D_t(1+g) = D_1(1+g)^{t-1}$, which allows to write

$$P_t = \mathbb{E}_t \left[\sum_{\tau=1}^{\infty} \frac{D_{t+\tau+1}(1+g)^{\tau-1}}{(1+R)^{\tau}} \right] = \mathbb{E}_t \left[\frac{D_{t+1}}{R-g} \right].$$

This is referred to as the Gordon (1962) growth model. Now assuming that price P_t is observed on the market and that R independent of D_{t+1} (the latter is generally a quite well ascertained quantity), the expected implied discount rate—thence the implied expected return on the security—can be solved for as

$$\mathbb{E}_t[R] = \frac{\mathbb{E}_t[D_{t+1}]}{P_t} + g.$$

Unfortunately, this model is very sensitive to inaccuracies in its inputs, and for this reason, so-called *residual income valuation* models (RIM) have been proposed that exploit the fundamental accounting *clean surplus relationship* linking the balance sheet and income statement

$$B_t = B_{t-1} + E_t - D_t, \tag{2.18}$$

*This model can be adapted to a similar *free cash flow* relationship for stocks that do not pay dividends.

†This hypothesis is valid, for instance, under the scenario where a business grows its earnings at a constant rate and maintains the same dividend payout ratio.

where B_t is the firm's book value per share at time t and E_t the earnings per share generated during period t. This states that the period-to-period variation in the firm's value is given by increases resulting from the period activities (net earnings) minus payments to shareholders (dividends) (Edwards and Bell 1961; Ohlson 1995). Define the "abnormal" earnings, assuming a constant discount factor R, as

$$E_t^a \triangleq E_t - R\,B_{t-1};$$

in this context, R can be interpreted as the required return on equity expected at the start of each period. This relationship, in conjunction with eq. (2.18), allows to write the dividends for period t as

$$D_t = E_t^a - B_t + (1+R)\,B_{t-1}.$$

Substituting in eq. (2.17), we obtain

$$\begin{aligned}
P_t &= \mathbb{E}_t\left[\frac{D_{t+1}}{1+R} + \frac{D_{t+2}}{(1+R)^2} + \cdots\right] \\
&= \mathbb{E}_t\left[\frac{E_{t+1}^a - B_{t+1} + (1+R)\,B_t}{1+R} + \frac{E_{t+2}^a - B_{t+2} + (1+R)\,B_{t+1}}{(1+R)^2} + \cdots\right] \\
&= B_t + \mathbb{E}_t\left[\sum_{\tau=1}^{\infty} \frac{E_{t+\tau}^a}{(1+R)^\tau}\right] \\
&= B_t + \mathbb{E}_t\left[\sum_{\tau=1}^{\infty} \frac{E_{t+\tau} - R\,B_{t+\tau-1}}{(1+R)^\tau}\right].
\end{aligned}$$

Under some assumptions, Philips (2003) derives the following expression for the expected returns

$$\mathbb{E}_t[R] = \frac{\mathbb{E}_t[E_{t+1}] - gB_t}{P_t} + g,$$

where P_t and B_t are readily available and E_{t+1} is often estimated by analysts that follow a stock.* The growth rate g can conservatively be taken as the growth of nominal GDP.† Claus and Thomas (2001) find relationships based on residual income valuations to be much less sensitive to errors than the Gordon model.

*Analyst forecasts of earnings have themselves long been subject to investigation, including the early work of Crichfield, Dyckman, and Lakonishok (1978) and Givoly and Lakonishok (1984), who generally find forecasts to improve as the earnings publication date approaches. More recently, Friesen and Weller (2006) consider a Bayesian framework in which analysts constantly revise their forecasts based on newly-revised information; in this context, the authors report strong evidence of biases, including overconfidence and cognitive dissonance biases.

†For firms whose capital structure consists of a mixture of equity and debt, this is indeed a very conservative assumption. The growth rate of nominal GDP would normally characterize the return on the firm's *assets*. In contrast, the return on *equity*—the quantity represented by g—would be magnified by the firm's financial leverage, i.e. its use of debt.

The topic of expected return forecasts is much richer than this brief overview can provide. In particular, we must omit treatment of a sizable literature on the information regarding the implied probability distribution of returns that option markets provide (e.g. Pan and Poteshman 2006; Aït-Sahalia and Brandt 2007). A review of several recently-proposed methodologies for forecasting expected returns appears in Satchell (2007).

2.1.8 Forecast Stability and Econometric Issues

A longstanding critique of Markowitz's mean-variance method of portfolio choice stems from the often-observed erratic nature of the optimal weights: unless expected returns are "perfectly matched" to the covariance matrix, it is frequent to arrive at *corner solutions* wherein a small number of assets get allocated most of the weight, with problem constraints strongly governing the obtained solution. It almost appears as if the theory's foundational goal of *efficient diversification of investment** somehow gets lost along the way. Moreover, the obtained solutions tend to be unstable, both cross-sectionally (small changes to the forecasts have a large impact on the weights) and over time (optimal portfolios often change drastically from one period to the next, leading to important costs due to turnover).

Michaud (1989) argues that extreme and unstable portfolio weights are inherent to mean-variance optimizers due to forecast estimation error: by virtue of mere statistical fluctuation, large positive (negative) weights are assigned to assets that have large positive (negative) estimation error in expected return and/or large negative (positive) error in variance. This arises because in the classical mean-variance paradigm, forecasts are totally disconnected from optimization: the former are "plugged into" the latter (hence the name *plug-in estimates*), and in a sense the optimizer "does not know" that the forecasts are but point estimates that also have an associated standard error. This led Michaud to his bon mot that *mean-variance optimizers act as statistical error maximizers.*

In addition to Michaud (1989), Jobson and Korkie (1980), Best and Grauer (1991) and Chopra and Ziemba (1993) study the impact of estimation uncertainty, where it is often observed to be much larger than that of asset risk itself. In particular, the plug-in estimates are found to be extremely unreliable, their performance dropping rapidly as the number of assets increases. This led to a variety of approaches to "robustify" the optimal portfolios, including shrinkage estimators, Bayesian approaches, resampling methods and robust optimization, summarized next. It should be noted that the practitioner's little-told secret of imposing optimization constraints, such as those reviewed in §2.1.6/p. 19, already serves to stabilize the portfolio by truncating extreme weights, and was confirmed by Frost and Savarino (1988) to generally improve performance. In this context, constraints can be interpreted as providing a *post hoc* regularization of the estimator (see §3.1/p. 69

*The subtitle in Markowitz's 1959 treatment of the subject.

for the associated machine learning theory), a point elaborated upon by Jagannathan and Ma (2003).

A very complete review of the literature on the econometrics of portfolio choice appears in Brandt (2004).

Shrinkage Estimators

It is known since Stein (1956) that biased estimators often have better finite-sample properties (lower sample variance) than unbiased ones.* In particular, consider estimating the mean of an N-dimensional ($N \geq 3$) multivariate normal distribution with *known covariance matrix* $\boldsymbol{\Sigma}$, subject to the quadratic loss function

$$L(\hat{\boldsymbol{\mu}}, \boldsymbol{\mu}) = (\hat{\boldsymbol{\mu}} - \boldsymbol{\mu})' \boldsymbol{\Sigma}^{-1} (\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}),$$

where $\boldsymbol{\mu}$ is the true mean. In this context, the usual sample mean $\hat{\boldsymbol{\mu}}$ is not the best estimator (James and Stein 1961). The James-Stein *shrinkage estimator*

$$\hat{\boldsymbol{\mu}}_{JS} = (1 - w)\hat{\boldsymbol{\mu}} + w\, \mu_0\, \boldsymbol{\iota}, \qquad 0 < w < 1,$$

exhibits a lower quadratic loss, where μ_0 is an arbitrary "common" constant and is called the shrinkage target. The optimal trade-off between bias and variance is achieved by

$$w^* = \min\left(1, \frac{(N-2)/T}{(\hat{\boldsymbol{\mu}} - \mu_0 \boldsymbol{\iota})' \boldsymbol{\Sigma}^{-1} (\hat{\boldsymbol{\mu}} - \mu_0 \boldsymbol{\iota})}\right).$$

More generally, shrinkage methods involve the combination of an unstructured estimator (with a large number of degrees of freedom and likely high sample variance) and a highly structured one (with a small number or even zero degrees of freedom). Jobson and Ratti (1979) and Jorion (1986) have studied them in a portfolio context, demonstrating that their benefits carries to the estimation of expected returns and obtain good performance of the resulting portfolios. Similarly, Frost and Savarino (1986) and Ledoit and Wolf (2004) apply them to the estimation of covariance matrices. Brandt (2004) suggests applying shrinkage estimation directly to the optimal portfolio weights, where the shrinkage target can be some *ex ante* reasonable weights such as $1/N$ or those of a benchmark portfolio.

Bayesian Approaches

In contrast to the "plug-in" approaches presented previously which sought to obtain the single best estimates of the next-period return mean and variance, a Bayesian or decision-theoretic approach would explicitly carry the estimation uncertainty to the optimization. Consider an explicit parameterization

*This *bias–variance trade-off* is related to the notion of capacity control which is studied in depth in machine learning; see §3.1/p. 69.

of the next-period return distribution, $P(\mathbf{R} \,|\, \boldsymbol{\theta})$, in terms of a parameter vector $\boldsymbol{\theta}$, allowing us to rewrite the expected utility maximization, eq. (2.11), as

$$\mathbf{w}^*(\boldsymbol{\theta}) = \arg\max_{\mathbf{w}} \int_{\mathbf{R}} U(\mathbf{w}'\mathbf{R}) \, \mathrm{d}P(\mathbf{R} \,|\, \boldsymbol{\theta}).$$

A Bayesian investor would not commit to a single choice of parameter vector $\boldsymbol{\theta}$, but would instead consider the posterior distribution of parameters, given by Bayes' rule as

$$P(\boldsymbol{\theta} \,|\, \mathcal{D}) = \frac{P(\mathcal{D} \,|\, \boldsymbol{\theta}) \, P_0(\boldsymbol{\theta})}{P(\mathcal{D})},$$

where $\mathcal{D}$ is some data (obviously only known up to before the start of the forecast period) and $P_0(\boldsymbol{\theta})$ is a (subjective) prior distribution on parameter values. The investor's subjective distribution of asset returns, given the data, is obtained by marginalizing out the parameters,

$$P(\mathbf{R} \,|\, \mathcal{D}) = \int_{\boldsymbol{\theta}} P(\mathbf{R} \,|\, \boldsymbol{\theta}) \, \mathrm{d}P(\boldsymbol{\theta} \,|\, \mathcal{D}),$$

yielding to reformulating the expected utility maximization problem for finding optimal portfolio weights as

$$\mathbf{w}^* = \arg\max_{\mathbf{w}} \int_{\boldsymbol{\theta}} \left[\int_{\mathbf{R}} U(\mathbf{w}'\mathbf{R}) \, \mathrm{d}P(\mathbf{R} \,|\, \boldsymbol{\theta}) \right] \mathrm{d}P(\boldsymbol{\theta} \,|\, \mathcal{D}).$$

This approach to portfolio choice was pioneered as early as the 1960's by Zellner and Chetty (1965) and further studied by Klein and Bawa (1976) and Brown (1978). More recently, the notion of a "learning investor" was revisited in the context of the increasing evidence on the (mild) predictability of returns in works by Kandel and Stambaugh (1996) and Barberis (2000); see §2.2.5/p. 55.

The Black-Litterman Model

A different path to Bayesian estimation relies on the implications of an underlying economic equilibrium model, which can serve to provide the "prior" in a portfolio choice context. This is embodied in the Black and Litterman (1992) model, widely used by practitioners. Our presentation of this model draws from Fabozzi, Focardi, and Kolm (2006).

Consider the expected-return relationship for asset i given by the CAPM (§2.1.7/p. 25),

$$\boldsymbol{\Pi}_i = \mathbb{E}[R_i - R_f] = \beta_i \mathbb{E}[R_M - R_f], \tag{2.19}$$

where β_i is obtained as a regression coefficient,

$$\beta_i = \frac{\mathrm{Cov}[R_i, R_M]}{\sigma_M^2},$$

with σ_M^2 the variance of the market portfolio. We shall denote by $\mathbf{w}_M$ the weights of the market portfolio, such that its return can be written as

$$R_M = \sum_{j=1}^{N} \mathbf{w}_{M,j} R_j.$$

Then eq. (2.19) can be rewritten as

$$\begin{aligned}
\mathbf{\Pi}_i &= \beta_i \mathbb{E}[R_M - R_f] \\
&= \frac{\operatorname{Cov}[R_i, R_M]}{\sigma_M^2} \mathbb{E}[R_M - R_f] \\
&= \frac{\operatorname{Cov}[R_i, \sum_{j=1}^{N} \mathbf{w}_{M,j} R_j]}{\sigma_M^2} \mathbb{E}[R_M - R_f] \\
&= \frac{\mathbb{E}[R_M - R_f]}{\sigma_M^2} \sum_{j=1}^{N} \mathbf{w}_{M,j} \operatorname{Cov}[R_i, R_j],
\end{aligned}$$

or in matrix form,

$$\mathbf{\Pi} = \delta \, \mathbf{\Sigma} \, \mathbf{w}_M \qquad \text{with} \quad \delta = \frac{\mathbb{E}[R_M - R_f]}{\sigma_M^2}.$$

Although the true expected asset returns $\boldsymbol{\mu}$ are unknown, we can posit that the equilibrium model provides a sensible approximation in the form of

$$\mathbf{\Pi} = \boldsymbol{\mu} + \boldsymbol{\varepsilon}_{\mathbf{\Pi}}, \qquad \boldsymbol{\varepsilon}_{\mathbf{\Pi}} \sim N(0, \tau \mathbf{\Sigma}), \tag{2.20}$$

where $\tau \ll 1$ is a small constant.* We can view $\boldsymbol{\varepsilon}_{\mathbf{\Pi}}$ as a "confidence interval" in which the true expected returns are approximated by the equilibrium model: a small τ implies a high confidence in the equilibrium estimates and vice versa.

Now suppose that the investor holds particular *views* on some assets or combinations of assets; examples are "the expected return of asset i will be x percent", or "asset j will outperform asset k by z percent". Each view has an attached *confidence* reflecting how strongly the investor believes them. We can formally express the K views as a vector $\mathbf{q} \in \mathbb{R}^K$,

$$\mathbf{q} = \mathbf{P}\boldsymbol{\mu} + \boldsymbol{\varepsilon}_{\mathbf{q}}, \qquad \boldsymbol{\varepsilon}_{\mathbf{q}} \sim N(0, \mathbf{\Omega}), \tag{2.21}$$

where $\mathbf{P}$ is a $K \times N$ matrix of view combinations and $\mathbf{\Omega}$ is a $K \times K$ matrix of view confidences. For example, in a universe of $N = 3$ assets, the investor may believe that

- Asset 1 will have a return of 1.5%.

*Which can be chosen by experimentation or sequential validation, see chapter 4; values in the neighborhood of 0.1–0.3 often give satisfactory results for U.S. equities.

- Asset 3 will outperform asset 2 by 4%.

This yields the following form for the views

$$\begin{bmatrix} 1.5\% \\ 4\% \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \\ \boldsymbol{\mu}_3 \end{bmatrix} + \begin{bmatrix} \boldsymbol{\varepsilon}_{\mathbf{q},1} \\ \boldsymbol{\varepsilon}_{\mathbf{q},2} \end{bmatrix},$$

for some view confidence matrix $\boldsymbol{\Omega}$, which is commonly diagonal. Both eq. (2.20) and (2.21) are expressed in terms of the unknown expected returns $\boldsymbol{\mu}$. The Black-Litterman model uses the *mixed estimator* of Theil and Goldberger (1961) to combine the information from two data sources—here the equilibrium model and the investor views—into a single posterior estimator. Start by "stacking" the two equations as follows,

$$\mathbf{y} = \mathbf{X}\boldsymbol{\mu} + \boldsymbol{\varepsilon}, \qquad \boldsymbol{\varepsilon} \sim N(0, \mathbf{V})$$

where

$$\mathbf{y} = \begin{bmatrix} \boldsymbol{\Pi} \\ \mathbf{q} \end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix} \mathbf{I}_N \\ \mathbf{P} \end{bmatrix}, \qquad \mathbf{V} = \begin{bmatrix} \tau\boldsymbol{\Sigma} & \\ & \boldsymbol{\Omega} \end{bmatrix}.$$

We can rely on a standard generalized least squares (GLS) estimator (Greene 2007) to arrive at the *Black-Litterman* estimator for expected returns,

$$\begin{aligned} \hat{\boldsymbol{\mu}}_{\mathrm{BL}} &= (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}^{-1}\mathbf{y} \\ &= \left(\begin{bmatrix} \mathbf{I}_N & \mathbf{P}' \end{bmatrix} \begin{bmatrix} (\tau\boldsymbol{\Sigma})^{-1} & \\ & \boldsymbol{\Omega}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{I}_N \\ \mathbf{P} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{I}_N & \mathbf{P}' \end{bmatrix} \begin{bmatrix} (\tau\boldsymbol{\Sigma})^{-1} & \\ & \boldsymbol{\Omega}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\Pi} \\ \mathbf{q} \end{bmatrix} \\ &= \left(\begin{bmatrix} \mathbf{I}_N & \mathbf{P}' \end{bmatrix} \begin{bmatrix} (\tau\boldsymbol{\Sigma})^{-1} \\ \boldsymbol{\Omega}^{-1}\mathbf{P} \end{bmatrix} \right)^{-1} \begin{bmatrix} \mathbf{I}_N & \mathbf{P}' \end{bmatrix} \begin{bmatrix} (\tau\boldsymbol{\Sigma})^{-1}\boldsymbol{\Pi} \\ \boldsymbol{\Omega}^{-1}\mathbf{q} \end{bmatrix} \\ &= \left[(\tau\boldsymbol{\Sigma})^{-1} + \mathbf{P}'\boldsymbol{\Omega}^{-1}\mathbf{P} \right]^{-1} \left[(\tau\boldsymbol{\Sigma})^{-1}\boldsymbol{\Pi} + \mathbf{P}'\boldsymbol{\Omega}^{-1}\mathbf{q} \right]. \end{aligned}$$

This estimator is then used with the standard mean-variance problem formulation, e.g. eq. (2.2) or eq. (2.7). Practical experience with this model, documenting the much greater stability of the resulting portfolio weights than would otherwise be obtained, is related in Bevan and Winkelmann (1998), Litterman (2003), and Fabozzi, Focardi, and Kolm (2006).

Portfolio Resampling

The Black-Litterman estimator still operates before portfolio optimization takes place; its benefits can be traced to a reduced "impedance mismatch" between the expected return estimator and the associated covariance matrix. In contrast, portfolio resampling techniques (Michaud 1998; Scherer 2002) attempt to make direct use of the forecast distribution of returns by repeatedly drawing a large number of (*expected-return* , *covariance-matrix*)

pairs, and for each computing an efficient frontier, namely a set of (*portfolio-return* , *portfolio-risk*) pairs, over some reasonable risk range. Then those efficient frontiers are averaged over all drawings, and the resulting frontier used to make an allocation decision. Markowitz and Usmen (2003) compare this approach to one similar to the Bayesian approach of p. 32 and observe a good performance of the resampling approach.

A practical limitation to the approach is with respect to *portfolio constraints*: in general, there is no guarantee that the averaged portfolio weights (after resampling) will obey the inequality constraints set in the original optimization problem. Also, due to the high number of optimization steps it requires, it is computationally expensive.

Robust Portfolio Allocation

In recent years, several reformulations of the mean-variance problem have received wide attention that attempt to incorporate estimation uncertainty within the optimization step—not "before", as for the Black-Litterman model, or "around" as for portfolio resampling. They are collectively known as *robust optimization* techniques, and are related to minimax estimators in decision theory.* Robust methods in mathematical programming were introduced by Ben-Tal and Nemirovski (1999) and further studied in a portfolio choice context by Goldfarb and Iyengar (2003) and Tütüncü and Koenig (2004) among others. Fabozzi, Kolm, Pachamanova, and Focardi (2007) provides a good survey of the current literature.

The starting point of these approaches is to consider the *uncertainty set* of the model parameters (the next-period expected returns and their covariances for a portfolio problem) and to ask: "what is the worst-case realization of model parameters that can arise?", and from there to maximize the utility of this worst-case outcome. Consider the simplest type of uncertainty region given in the form of "box" intervals

$$\mathcal{U} = \{(\boldsymbol{\mu}, \boldsymbol{\Sigma}) : \boldsymbol{\mu}_L < \boldsymbol{\mu} < \boldsymbol{\mu}_U, \boldsymbol{\Sigma}_L < \boldsymbol{\Sigma} < \boldsymbol{\Sigma}_U, \boldsymbol{\Sigma} \succ 0\},$$

where in this context the $<$ operator should be interpreted elementwise for both vectors and matrices.

The robust portfolio problem with quadratic utility is expressed as

$$\max_{\mathbf{w}} \left\{ \min_{(\boldsymbol{\mu},\boldsymbol{\Sigma})\in\mathcal{U}} \boldsymbol{\mu}'\mathbf{w} - \lambda \mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w} \right\}$$

which for the above form of the uncertainty region separates out as

$$\max_{\mathbf{w}} \left\{ \min_{\boldsymbol{\mu}\in\mathcal{U}^{\boldsymbol{\mu}}} \boldsymbol{\mu}'\mathbf{w} + \max_{\boldsymbol{\Sigma}\in\mathcal{U}^{\boldsymbol{\Sigma}}} \lambda \mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w} \right\}.$$

*Robust optimization should not be confused with *robust estimation* in statistics, devoted to establishing the properties of outlier-resistant estimators.

This can be expressed as a saddle-point problem and solved in polynomial time (Halldórsson and Tütüncü 2003). Simpler results can be obtained by considering other types of uncertainty sets; for instance, when only uncertainty in expected returns is considered, the box constraints reduce to a quadratic program of nearly the same complexity as the original mean-variance problem; similarly, an ellipsoidal constraint set yields a second-order cone program (SOCP), which is efficiently solved by interior-point methods (Boyd and Vandenberghe 2004). More recently, Bertsimas and Pachamanova (2008) studied a number of robust optimization approaches to the multiperiod portfolio problem (see next section) in the presence of transaction costs; in particular, they advocate linear formulations that yield significant computational savings.

Portfolio Robustness: a Synthesis?

In light of the large variety of proposed methods for improving the performance of mean-variance allocation, one may wonder if a particular method turns out to be "best". To the author's knowledge, a systematic comparison between all of the approaches presented in this section has yet to be published. However, an element of insight has recently been provided by DeMiguel, Garlappi, and Uppal (2009), who compare 14 different models on a number of datasets (including U.S. and world equity markets) on three criteria: the out-of-sample Sharpe ratio, certainty equivalent return (from the perspective of a mean-variance investor) and portfolio turnover. On these measures, it is found that *none of the "sophisticated" models consistently beat the naïve* $1/N$ *benchmark* (uniform portfolio weights), *out of sample*. These results suggest that, for the models considered, estimation error still largely dominates any gains obtained from "optimal" diversification.

2.2 Multiperiod Problems

Multiperiod problems consider the more general case where an investor makes a *sequence* of decisions, each possibly impacting the following ones. The objective is to find, at each period, the allocation decision that take into consideration a future changing opportunity set (i.e. availability of assets and their risk–return characteristics), the remaining investment horizon, eventual transaction costs, and other constraints such as the desire for intermediate consumption, minimization of tax impact, or the influx of additional capital due to labor income. These decisions, in general, are not identical to those obtained under the myopic (one-period) case, although they can be under specific assumptions (see §2.2.3/p. 47); more often, we shall see that the optimal solution is constructed from the myopic one as a starting point which is perturbed by a *hedging demands* term to account

for "the future". This term makes the obtained portfolio policies differ from iterated single-period ones.

Although Markowitz (1959) discussed the use of dynamic programming to solve the sequential optimal portfolio choice problem (using a time-separable log-utility of consumption as the objective function), he disregarded its more systematic application as computationally unfeasible:

> "For the actual choice of portfolio, however, the dynamic programming techniques cannot be used. They require too much both from man and machine: 1. From the investor they require a utility function $U(C_1, C_2, \ldots, C_t)$. [...] It is no small task to derive a reasonable single period utility function. [...] To attempt to derive a representative utility function for consumption over time, if feasible at all, is nothing short of a major research project. 2. Even with the simplest of utility functions, the requirements for the dynamic programming computation are far beyond economic justification." (p. 278)

Just as the single-period problem, the multiperiod generalization has a rich history, albeit a more academic one.* Samuelson (1969) and Merton (1969) are generally credited with posing the general multiperiod consumption and investment problem, Samuelson in discrete time (§2.2.1/p. 38) and Merton in continuous time (§2.2.2/p. 44), although Mossin (1968) had previously studied the multiperiod portfolio choice problem (without a consumption aspect). Earlier closely related work includes Tobin (1965) and Phelps (1962) who considers the lifetime utility associated with a consumption history.

After introducing these foundations, we review classical results on the structure of optimal policies (§2.2.3/p. 47) including a discussion of the optimality conditions of the myopic policy. They are seen to be strongly impacted by the expected evolution of the *investment opportunity set*, namely the risk-reward characteristics of available assets. We give passing mention of an elegant alternative to dynamic programming based on the martingale formulation (§2.2.4/p. 49) and models that explicitly incorporate consideration of investor learning behavior (§2.2.5/p. 55). We end this section by giving pointers to common extensions (§2.2.6/p. 55) that have been proposed.

2.2.1 The Discrete-Time Case

Consider the problem where at each time-step $t = 0, 1, 2, \ldots, T-1$ the investor makes a portfolio choice $\mathbf{w}_t$ wherein he tries to intertemporally maximize the expected utility of wealth at the final time T, $U(W(T))$, given

*To the author's knowledge, multiperiod optimization has yet to be used in the day-to-day management of an institutional portfolio. This, perhaps, can be attributed to the perceived small gains of the approach compared to its complexity and the remaining inevitable overall portfolio risk.

a current wealth $W_t \in \mathbb{R}$,

$$\max_{\mathbf{w}_0, \mathbf{w}_1, \ldots, \mathbf{w}_{T-1}} \mathbb{E}_t\big[U(W(T))\big], \tag{2.22}$$

subject the the *budget constraint*

$$W_{t+1} = W_t(1 + \mathbf{w}_t'\mathbf{R}_{t+1} + (1 - \mathbf{w}_t'\boldsymbol{\iota})R_{f,t}), \qquad W_0 \text{ given.} \tag{2.23}$$

This constraint describes the dynamics of wealth, specifying that the total relative return experienced during period $t+1$ arises from the allocation $\mathbf{w}_t$ to risky assets and the remainder $(1 - \mathbf{w}_t'\boldsymbol{\iota})$ from the risk-free asset; note that the latter quantity can be negative, in which case the investor borrows at the risk-free rate.* We also require wealth to be always nonnegative, $W_t \geq 0$. Given a sequence of decisions $\{\mathbf{w}_\tau\}_{\tau=t}^{T-1}$, it is useful to observe that the terminal wealth W_T can be written as a function of current wealth W_t,

$$W_T = W_t \prod_{\tau=t}^{T-1} \big(1 + \mathbf{w}_\tau'\mathbf{R}_{\tau+1} + (1 - \mathbf{w}_\tau'\boldsymbol{\iota})R_{f,\tau}\big). \tag{2.24}$$

Consistent with a formulation by dynamic programming (Bellman 1957; Bertsekas 2005), it is convenient to express the expected terminal wealth in terms of a *value function*, varying according to the current time,† current wealth W_t and other state variables $\mathbf{z}_t \in \mathbb{R}^K, K < \infty$,

$$\begin{aligned} V(t, W_t, \mathbf{z}_t) &= \max_{\{\mathbf{w}_u\}_{u=t}^{T-1}} \mathbb{E}_t\Big[U(W_T)\Big] \\ &= \max_{\mathbf{w}_t} \mathbb{E}_t\left[\max_{\{\mathbf{w}_u\}_{u=t+1}^{T-1}} \mathbb{E}_{t+1}\Big[U(W_T)\Big]\right] \qquad (2.25) \\ &= \max_{\mathbf{w}_t} \mathbb{E}_t\Big[V(t+1, W_{t+1}, \mathbf{z}_{t+1})\Big], \qquad (2.26) \end{aligned}$$

subject to the budget constraint (2.23) and the recursive base case

$$V(T, W_T, \mathbf{z}_T) = U(W_T).$$

The expectations at time t, above, are taken with respect to the joint distribution of asset returns and next state, conditional on the information available at time t, $P(\mathbf{R}_{t+1}, \mathbf{z}_{t+1} \,|\, \mathcal{F}_t)$. For our purposes, it shall be sufficient to assume a first-order Markov process for this, such that

$$P(\mathbf{R}_{t+1}, \mathbf{z}_{t+1} \,|\, \mathcal{F}_t) = P(\mathbf{R}_{t+1}, \mathbf{z}_{t+1} \,|\, \mathbf{R}_t, \mathbf{z}_t);$$

*A more complex constraint can account for differing lending and borrowing rates.

†Regarding notation, many treatments of finite-horizon discrete-time dynamic programming (e.g. Bertsekas 2005) simply consider a set of value functions indexed by the current time-step, V_t; here we specifically include time as an explicit variable to preserve notational consistency with the continuous-time treatment in §2.2.2/p. 44.

this assumption is not overly restrictive in practice since $\mathbf{z}_t$ can contain (a finite number of) lagged values of relevant variables.

In what follows, we shall use the notation $f_i(\cdot)$ to denote the partial derivative of function f with respect to the i-th argument, e.g.

$$V_2(t', W', \mathbf{z}') \triangleq \left.\frac{\partial V}{\partial W}\right|_{t=t', W=W', \mathbf{z}=\mathbf{z}'}.$$

From eq. (2.26), the first-order conditions for optimality at each time t are obtained as

$$\begin{aligned} 0 &= \mathbb{E}\left[V_2(t+1, W_{t+1}, \mathbf{z}_{t+1})\frac{\partial W_{t+1}}{\partial \mathbf{w}_t}\right] \\ &= \mathbb{E}\left[V_2\Big(t+1, W_t\big(1 + \mathbf{w}_t'\mathbf{R}_{t+1} + (1 - \mathbf{w}'\boldsymbol{\iota})R_{f,t}\big), \mathbf{z}_{t+1}\Big)\mathbf{R}_{t+1}\right], \end{aligned} \tag{2.27}$$

These optimality conditions assume that the state variable $\mathbf{z}_{t+1}$ is not impacted by the decision $\mathbf{w}_t$.* The second-order conditions are satisfied if the utility function is concave.

Mossin (1968) studied this problem under the assumption of independence of returns across time-steps, no transaction costs and no intermediate consumption. He derived conditions for which the myopic policy can be optimal (§2.2.3/p. 47). Samuelson (1969) studied the related problem in which the investor derives utility from intermediate consumption and tries to maximize both the discounted utility of the consumption stream and the utility of terminal ("bequeathed") wealth.†

Power Utility

In general, (2.27) can only be solved numerically. However, some analytic progress can be achieved in the case of the *power utility*,

$$U(W) = \begin{cases} \frac{W^{1-\alpha}}{1-\alpha}, & \alpha \neq 1 \\ \ln W, & \text{otherwise,} \end{cases}$$

where α is a coefficient of relative risk aversion. This is an example of a constant relative risk aversion (CRRA) utility function, discussed in §2.1.4/p. 14. In this case (assuming, for simplicity, $\alpha \neq 1$), substituting in eq. (2.25), we

*This would disregard, for instance, the market impact of trading for large market players. Kissell and Glantz (2003) consider market impact at length.

†Samuelson imposes the "greedy granny" condition, i.e. a zero-bequest requirement as a boundary condition.

obtain

$$
\begin{aligned}
V(t, W_t, \mathbf{z}_t) &= \max_{\mathbf{w}_t} \mathbb{E}_t \left[\max_{\{\mathbf{w}_\tau\}_{\tau=t+1}^{T-1}} \mathbb{E}_{t+1} \left[\frac{W_T^{1-\alpha}}{1-\alpha} \right] \right] \\
&= \max_{\mathbf{w}_t} \mathbb{E}_t \left[\max_{\{\mathbf{w}_\tau\}_{\tau=t+1}^{T-1}} \mathbb{E}_{t+1} \left[\frac{\left(W_t \prod_{\tau=t}^{T-1} \left(1 + \mathbf{w}_\tau' \mathbf{R}_{\tau+1} + (1 - \mathbf{w}_\tau' \boldsymbol{\iota}) R_{f,\tau} \right) \right)^{1-\alpha}}{1-\alpha} \right] \right] \\
&= \max_{\mathbf{w}_t} \mathbb{E}_t \Bigg[\underbrace{\frac{\left(W_t \left(1 + \mathbf{w}_t' \mathbf{R}_{t+1} + (1 - \mathbf{w}_t' \boldsymbol{\iota}) R_{f,t} \right) \right)^{1-\alpha}}{1-\alpha}}_{U(W_{t+1})} \times \\
&\qquad \max_{\{\mathbf{w}_\tau\}_{\tau=t+1}^{T-1}} \mathbb{E}_{t+1} \Bigg[\underbrace{\left(\prod_{\tau=t+1}^{T-1} \left(1 + \mathbf{w}_\tau' \mathbf{R}_{\tau+1} + (1 - \mathbf{w}_\tau' \boldsymbol{\iota}) R_{f,\tau} \right) \right)^{1-\alpha}}_{\psi(t+1, \mathbf{z}_{t+1})} \Bigg] \Bigg].
\end{aligned}
$$

In the next-to-last expression, the specific form of the power utility allows W_t to be factored out of the maximizations since it is not impacted by the decision variables $\{\mathbf{w}_u\}_{u=t+1}^{T-1}$. Hence, the last expression shows that the value function factors out into two parts: a first one, that depends on future wealth, and equal to the utility of next-time-step wealth, $U(W_{t+1})$, and a second one that only depends on remaining time horizon and future state variables $\mathbf{z}_{t+1}$, but not future wealth. This can further be reduced by writing

$$
V(t, W_t, \mathbf{z}_t) = \frac{(W_t)^{1-\alpha}}{1-\alpha} \psi(t, \mathbf{z}_t)
$$

where $\psi(t, \mathbf{z}_t)$ satisfies Bellman's equation in a smaller state space,

$$
\psi(t, \mathbf{z}_t) = \max_{\mathbf{w}_t} \mathbb{E}_t \left[\left(1 + \mathbf{w}_t' \mathbf{R}_{t+1} + (1 - \mathbf{w}_t' \boldsymbol{\iota}) R_{f,t} \right)^{1-\alpha} \psi(t+1, \mathbf{z}_{t+1}) \right], \quad (2.28)
$$

with recursive base case

$$
\psi(T, \cdot) = 1. \quad (2.29)
$$

If the returns are IID*, the above joint expectation between returns and state variables splits out as

*Independent and Identically Distributed.

$$
\psi(t, \mathbf{z}_t) = \max_{\mathbf{w}_t} \left\{ \mathbb{E}_t \left[\left(1 + \mathbf{w}_t' \mathbf{R}_{t+1} + (1 - \mathbf{w}_t' \boldsymbol{\iota}) R_{f,t} \right)^{1-\alpha} \right] \right\} \mathbb{E}_t \left[\psi(t+1, \mathbf{z}_{t+1}) \right], \quad (2.30)
$$

where it is readily seen that the optimal portfolio weights at each time-step are independent of the state variables and remaining time horizon, thence must be constant. Put differently, for IID returns (and power utility), there is no difference between the dynamic and myopic portfolios; this property is revisited in §2.2.3/p. 47.

Numerical Example

Given a model of the conditional return distribution, the Bellman equations (2.28)–(2.29) can be solved numerically. For a power-utility investor, on a two-asset problem—shifting wealth between a riskless bond and a single risky asset—and using the generative model of eq. (2.16), conditioning excess return on dividend yield, some instructive results appear in Fig. 2.4.*

The left panel shows the fraction of wealth invested in the risky asset as a function of the initial dividend yield—in effect at the time of making the forecast—and various investor risk aversion levels, for the single-period problem (horizon=1). The plot clearly shows that lower-aversion individuals shift their allocation very rapidly for increasing forecasted returns (as indicated by the dividend yield) in the risky asset: this corresponds to an increased propension for *market timing* as risk aversion decreases.

The right panel illustrates the *horizon effects* that arise in the presence of return forecastability (but are, as noted above, absent when returns are assumed IID). It shows the allocation to the risky asset as a function of investment horizon, for various *initial* (i.e. first-period) dividend yields and a constant risk aversion $\alpha = 5$. No matter how bleak the immediate prospects for risky returns are (i.e. low dividend yield), a long-horizon investor allocates more to the risky asset than a short-horizon one, since the returns will eventually revert to their unconditional mean over a long period; however, in the model of eq. (2.28), this mean-reversion takes time due to the high autocorrelation in the (log) dividend yield.

A more complete picture of the optimal policy, and associated value function, is given in Fig. 2.5.

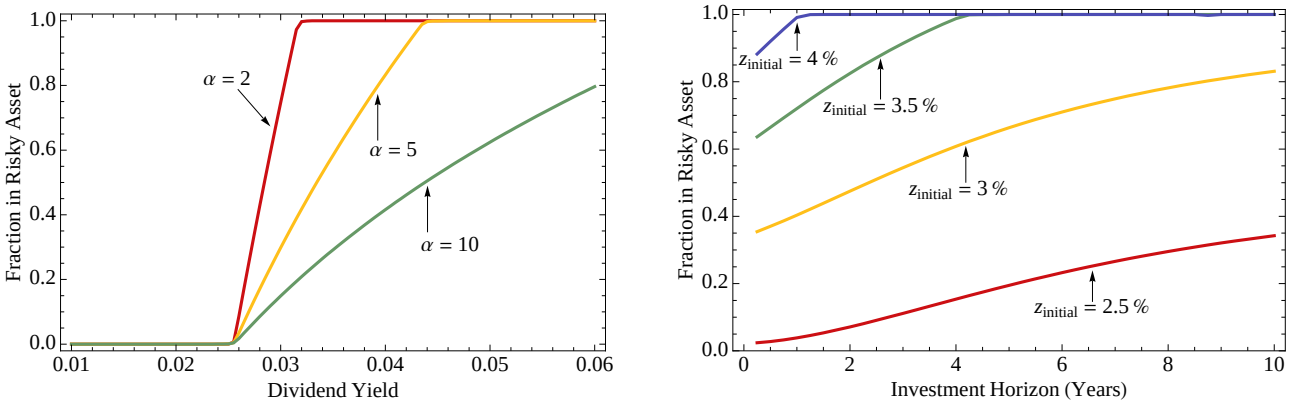
The Mean-Variance Multiperiod Criterion

Surprisingly, it has been only relatively recently that a multiperiod analog to the mean-variance problem received a thorough solution in the discrete-time case.† Li and Ng (2000) analyzed various formulations of the maximization of terminal quadratic utility under several hypotheses, provided explicit solutions in simplifying cases, and derived analytical expressions for the multiperiod mean-variance efficient frontier (a concept that had, until that point, received no attention in the multiperiod case).

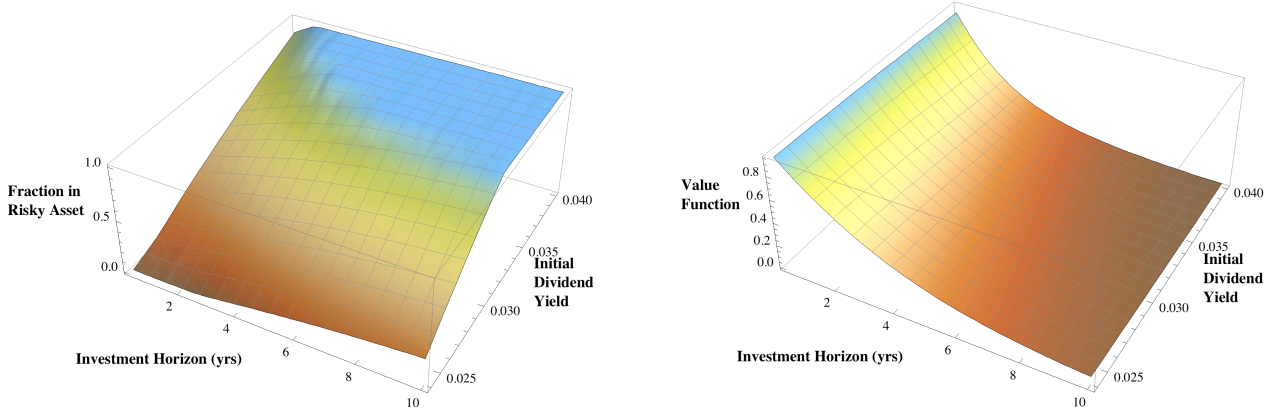
More specifically, considering the N-risky-asset case as previously, the form of the mean-variance optimization problem follows the minimum-variance formulation (2.2)–(2.4) or the utility-maximization formulation (2.7)–(2.8), with the exception that the objective function is expressed in terms of

*The simulations are carried out by estimating the expectation in (2.28) by Monte Carlo sampling with 2500 trajectories. Maximization is performed by numerical optimization using Mathematica 6's built-in `NMaximize` function for constrained maximization without necessitating the availability of gradients.

†In continuous time, the problem was solved by Korn and Trautmann (1995) and Zhou (2000).



▲ **Figure 2.4.** ***Left:*** *Fraction of wealth invested in the risky asset for the two-asset problem as a function of the initial dividend yield, for an investment horizon of one period (one quarter, in this case) and various investor risk aversion levels (α).* ***Right:*** *Fraction of wealth invested in the risky asset as a function of the investment horizon, for various initial dividend yields and constant risk aversion $\alpha = 5$.*



▲ **Figure 2.5.** ***Left:*** *Optimal policy (fraction of capital invested in the risky asset) as a function of time-to-maturity (years) and initial dividend yield, for an investor with a constant risk-aversion $\alpha = 5$.* ***Right:*** *Value function under the same conditions.*

terminal *wealth* instead of portfolio relative return. For instance, the utility formulation takes the form

$$\max_{\{\mathbf{w}_t\}_{t=1}^{T-1}} \quad \mathbb{E}[W_T] - \lambda \operatorname{Var}[W_T] \tag{2.31}$$

$$\text{subject to} \quad W_{t+1} = W_t \mathbf{w}_t'(1 + \mathbf{R}_{t+1}) \tag{2.32}$$

$$\boldsymbol{\iota}'\mathbf{w} = 1, \tag{2.33}$$

where the initial wealth W_0 is given. The derivation of optimal solutions is complicated by the fact that the objective is not time-separable (in the dynamic programming sense), but analytical solutions exist if asset returns are assumed independent between periods; they do not need to be identically distributed, provided that all future return means and covariance matrices are known ahead of time. Obviously, the estimation methodology of §2.1.7/p. 24 can be put to bear for this task. Moreover, §2.2.5/p. 55 connect these mildly unrealistic assumptions with the fact that in a multiperiod setting, the optimal policy depends on the fact that we *expect* to learn more about the asset return distribution in the future.

Leippold, Trojani, and Vanini (2004) provided an interpretation of the solution to the multiperiod mean-variance problem in terms of an orthogonal set of basis strategies, each with a clear economic interpretation. They use this analysis to provide analytical solutions to portfolios consisting of both assets and liabilities. More recently, Cvitanić, Lazrak, and Wang (2008) connect this problem to a specific case of multiperiod Sharpe ratio maximization.

2.2.2 The Continuous-Time Case

The continuous-time analysis is due to Merton (1969, 1971) and illustrates the analytical tractability of the approach; Merton's seminal papers consider the joint optimization of investment and consumption decisions, including a number of variations including the effect of wage income and alternative stochastic processes. Our succinct exposition draws from Brandt (2004) and for simplicity, only considers the maximization of the terminal utility of wealth—disregarding intermediate consumption—and assumes that all assets are driven by log-diffusion processes (geometric Brownian motion) and can be traded continuously without friction (transaction costs, taxes) or consideration of background risk (general economic downturn, unemployment risk).* Despite this simplified setting, the results obtained are sufficiently illuminating to convey noteworthy intuition about the structure of the optimal multiperiod portfolio choice.

In continuous time, the problem formulation is identical to the discrete-time objective (2.22), except that instead of making a discrete set of deci-

*See §2.2.6/p. 55 for references to the many extensions that have been proposed to address these restrictions. See Merton (1990) and Duffie (2001) for more formal treatments of the material in this section.

sions, a continuous allocation trajectory must be found, subject to a continuous-time budget constraint. We shall assume, for $0 \leq t < T, t \in \mathbb{R}$, that the N risky asset prices $\mathbf{P}_t$ and K-dimensional state vector $\mathbf{z}_t$ evolve jointly according to correlated Itô vector processes,*

$$\frac{\mathrm{d}\mathbf{P}_t}{\mathbf{P}_t} = (\boldsymbol{\mu}^{\mathbf{P}}(\mathbf{z}_t, t) + r_f)\mathrm{d}t + \mathbf{D}^{\mathbf{P}}(\mathbf{z}_t, t)\mathrm{d}\mathbf{B}_t^{\mathbf{P}} \tag{2.34}$$

$$\mathrm{d}\mathbf{z}_t = \boldsymbol{\mu}^{\mathbf{z}}(\mathbf{z}_t, t)\mathrm{d}t + \mathbf{D}^{\mathbf{z}}(\mathbf{z}_t, t)\mathrm{d}\mathbf{B}_t^{\mathbf{z}} \tag{2.35}$$

subject to the budget constraint

$$\frac{\mathrm{d}W_t}{W_t} = \big(\mathbf{w}_t'\boldsymbol{\mu}_t^{\mathbf{P}} + r_f\big)\mathrm{d}t + \mathbf{w}_t'\mathbf{D}_t^{\mathbf{P}}\mathrm{d}\mathbf{B}_t^{\mathbf{P}} \tag{2.36}$$

where $\boldsymbol{\mu}^{\mathbf{P}}(\mathbf{z}_t, t)$ is the conditional mean excess return of the assets, $\mathbf{D}^{\mathbf{P}}(\mathbf{z}_t, t)$ is the conditional $N \times N$ price process diffusion matrix, $\boldsymbol{\mu}^{\mathbf{z}}(\mathbf{z}_t, t)$ is the conditional drift of the state variables, and $\mathbf{D}^{\mathbf{z}}(\mathbf{z}_t, t)$ is the conditional $K \times K$ state process diffusion matrix. For readability, we drop the explicit conditioning on $(\mathbf{z}_t, t)$ in what follows, and use a simple t subscript for the previous quantities. The diffusion matrices $\mathbf{D}_t^{\mathbf{P}}$ and $\mathbf{D}_t^{\mathbf{z}}$ respectively induce covariance matrices $\boldsymbol{\Sigma}_t^{\mathbf{P}}$ and $\boldsymbol{\Sigma}_t^{\mathbf{z}}$ within the price process $\mathbf{P}_t$ and state process $\mathbf{z}_t$. Furthermore, we assume that the underlying Brownian processes $\mathbf{B}_t^{\mathbf{P}}$ and $\mathbf{B}_t^{\mathbf{z}}$ are related by a time-varying $N \times K$ correlation matrix $\boldsymbol{\rho}_t$.† The notation $\mathrm{d}\mathbf{P}_t/\mathbf{P}_t$ should be interpreted as elementwise differentiation.

To derive the continuous-time Bellman equation, we can proceed informally as follows (see Merton (1971) for a more complete treatment). We obtain it as the limit as $\Delta t \to 0$ of the discrete-time Bellman equation (2.26). First the equation is rewritten as

$$0 = \max_{\mathbf{w}_t} \mathbb{E}_t\big[V(t+1, W_{t+1}, \mathbf{z}_{t+1}) - V(t, W_t, \mathbf{z}_t)\big]$$

and we replace the transition to the "next" time-step by an interval Δt,

$$0 = \max_{\mathbf{w}_t} \mathbb{E}_t\big[V(t+\Delta t, W_{t+\Delta t}, \mathbf{z}_{t+\Delta t}) - V(t, W_t, \mathbf{z}_t)\big],$$

which yields, in the limit of $\Delta t \to 0$,

$$0 = \max_{\mathbf{w}_t} \mathbb{E}_t\big[\mathrm{d}V(t, W_t, \mathbf{z}_t)\big]. \tag{2.37}$$

We can then mechanically apply Itô's lemma (A.20) (see p. 299) to the value function to derive (for notational convenience, V is used in place of

*This section assumes some familiarity with stochastic differential equations; see §A.3/p. 298 for a review of Itô's lemma, used in the derivations to follow.

†Note that the elements *within* both $\mathrm{d}\mathbf{B}_t^{\mathbf{P}}$ and $\mathrm{d}\mathbf{B}_t^{\mathbf{z}}$ are uncorrelated; all the "inner" correlation structure within the processes $\mathbf{P}_t$ and $\mathbf{z}_t$ is induced though the off-diagonal terms in the diffusion matrices $\mathbf{D}_t^{\mathbf{P}}$ and $\mathbf{D}_t^{\mathbf{z}}$.

$V(t, W_t, \mathbf{z}_t)$ when no confusion is possible)

$$\begin{aligned} \mathrm{d}V = V_1 \mathrm{d}t + V_2 \mathrm{d}W_t + V_3 \mathrm{d}\mathbf{z}_t + \\ \frac{1}{2} V_{2,2}\, \mathrm{d}W_t^2 + V_{2,3}\, \mathrm{d}W_t\, \mathrm{d}\mathbf{z}_t + \frac{1}{2} V_{3,3}\, \mathrm{d}\mathbf{z}_t^2, \end{aligned}$$

where, as previously, the notation V_i denotes the partial derivative of V with respect to the i-th argument (note that V_1 and V_2 are scalars, whereas V_3 is a K-vector). Substituting $\mathrm{d}\mathbf{z}_t$ and $\mathrm{d}W_t$ respectively from eq. (2.35) and (2.36) and applying the usual rules for the product of differentials* we obtain

$$\begin{aligned} \mathrm{d}V = V_1 \mathrm{d}t + V_2 W_t (\mathbf{w}_t' \boldsymbol{\mu}_t^{\mathbf{P}} + r_f)\mathrm{d}t + V_2 W_t \mathbf{w}_t' \mathbf{D}_t^{\mathbf{P}} \mathrm{d}\mathbf{B}_t^{\mathbf{P}} + \\ V_3' \boldsymbol{\mu}_t^{\mathbf{z}} \mathrm{d}t + V_3' \mathbf{D}_t^{\mathbf{z}} \mathrm{d}\mathbf{B}_t^{\mathbf{z}} + \frac{1}{2} V_{2,2} W_t^2 \mathbf{w}_t' \mathbf{D}_t^{\mathbf{P}} I_N \mathbf{D}_t^{\mathbf{P}\prime} \mathbf{w}_t \mathrm{d}t + \\ W \mathbf{w}_t' \mathbf{D}_t^{\mathbf{P}} \boldsymbol{\rho}_t' \mathbf{D}^{\mathbf{z}\prime} V_{2,3} \mathrm{d}t + \frac{1}{2} \mathrm{tr}\big[\mathbf{D}_t^{\mathbf{z}} I_K \mathbf{D}_t^{\mathbf{z}\prime} V_{3,3}\big] \mathrm{d}t. \end{aligned}$$

Taking expectations, substituting back into eq. (2.37), dividing left and right sides by $\mathrm{d}t$, and rearranging terms, we obtain the continuous-time Bellman equation,

$$\begin{aligned} 0 = \max_{\mathbf{w}_t} \Big[V_1 + W_t (\mathbf{w}_t' \boldsymbol{\mu}_t^{\mathbf{P}} + r_f) V_2 + \boldsymbol{\mu}_t^{\mathbf{z}\prime} V_3 + \\ \frac{1}{2} W_t^2 \mathbf{w}_t' \boldsymbol{\Sigma}_t^{\mathbf{P}} \mathbf{w}_t V_{2,2} + W_t \mathbf{w}_t' \mathbf{D}_t^{\mathbf{P}} \boldsymbol{\rho}_t' \mathbf{D}_t^{\mathbf{z}\prime} V_{2,3} + \frac{1}{2} \mathrm{tr}\big[\boldsymbol{\Sigma}_t^{\mathbf{z}} V_{3,3}\big] \Big] \end{aligned} \tag{2.38}$$

subject to terminal conditions $V(T, W_T, \mathbf{z}_T) = U(W_T)$. The first-order conditions for optimality are obtained as a stationary point of eq. (2.38) with respect to $\mathbf{w}_t$,

$$\boldsymbol{\mu}_t^{\mathbf{P}} V_2 + W_t V_{2,2} \boldsymbol{\Sigma}_t^{\mathbf{P}} \mathbf{w}_t + \mathbf{D}_t^{\mathbf{P}} \boldsymbol{\rho}_t' \mathbf{D}_t^{\mathbf{z}\prime} V_{2,3} = 0,$$

which can explicitly be solved for the optimal portfolio weights

$$\begin{aligned} \mathbf{w}_t^* = & \underbrace{-\frac{V_2}{W_t V_{2,2}} \big(\boldsymbol{\Sigma}_t^{\mathbf{P}}\big)^{-1} \boldsymbol{\mu}_t^{\mathbf{P}}}_{\text{Myopic Portfolio}} \\ & \underbrace{- \frac{V_2}{W_t V_{2,2}} \frac{V_{2,3}}{V_2} \big(\boldsymbol{\Sigma}_t^{\mathbf{P}}\big)^{-1} \mathbf{D}_t^{\mathbf{P}} \boldsymbol{\rho}_t' \mathbf{D}_t^{\mathbf{z}\prime}}_{\text{Hedging Demands}} \end{aligned} \tag{2.39}$$

The "myopic portfolio" term corresponds to the solution of the one-period problem and is equivalent to eq. (2.10). The factor $-V_2/(W_t V_{2,2})$ represents

*Namely,

$$\begin{aligned} &(\mathrm{d}t)^2 = 0, && \mathrm{d}t\, (\mathrm{d}\mathbf{B}_t^{\{\mathbf{P},\mathbf{z}\}})_i = 0, \\ &(\mathrm{d}\mathbf{B}_t^{\mathbf{P}})_i (\mathrm{d}\mathbf{B}_t^{\mathbf{P}})_j = \delta_{i,j} \mathrm{d}t, && (\mathrm{d}\mathbf{B}_t^{\mathbf{z}})_i (\mathrm{d}\mathbf{B}_t^{\mathbf{z}})_j = \delta_{i,j} \mathrm{d}t, && (\mathrm{d}\mathbf{B}_t^{\mathbf{P}})_i (\mathrm{d}\mathbf{B}_t^{\mathbf{z}})_j = \boldsymbol{\rho}_{i,j} \mathrm{d}t. \end{aligned}$$

the investor's relative risk tolerance (reciprocal of the relative risk aversion). The "hedging demands" term corresponds to an additional demand for risky assets resulting from changes in the investment opportunity set. It depends on the following factors:

- Non-constant state variables (the $\mathbf{D}_t^{\mathbf{z}}$ matrix must be non-zero);
- Correlation between state variables and risky-asset prices ($\boldsymbol{\rho}_t$);
- How strongly the changes in state variables $\mathbf{z}_t$ affect the utility of wealth ($V_{2,3}$ factor).

If any of those factors is zero, hedging demands disappear and only the myopic portfolio remains. Hence, the presence of hedging demands depends on the ability of state variables to capture (instantaneous) changes in the asset price process. The marginal utility of wealth with respect to the state variables ($V_{2,3}/V_2$) chooses the appropriate tradeoff between the myopic and hedging terms. Brandt (2004) offers the following interpretation on the relationship between state variable and asset price processes: "The projection $\left((\boldsymbol{\Sigma}_t^{\mathbf{P}})^{-1}\mathbf{D}_t^{\mathbf{P}}\boldsymbol{\rho}_t'\mathbf{D}_t^{\mathbf{z}\prime}\right)$ delivers the weights of K portfolios that are maximally correlated with the state variable innovations and the derivatives of marginal utility with respect to the state variables measure how important each of these state variables is to the investor. Intuitively, the investor takes positions in each of the maximally correlated portfolios to partially hedge against undesirable innovations in the state variables."

Merton (1973) presents a very elegant version of the CAPM wherein all investors are assumed to be intertemporal maximizers (as opposed to single-period Markowitz maximizers as in the original CAPM; see §2.1.7/p. 25) and considers equilibrium relations among expected returns; as such, he derives solutions for the price of risk that are quite different from the CAPM's market β. In particular, it is shown that even though a risky asset exhibits no "systematic" (or market) risk, it can earn a return different from the risk-free rate due to the hedging demands introduced above.

2.2.3 Structure of Optimal Solutions

A major concern in the classical analyses is with respect to the *structure of optimal solutions*. In continuous time, the problem can be solved analytically only for a handful of special cases; for instance, in the case of the power utility, one can assume—just as for discrete-time—a separable solution of the form

$$V(t, W_t, \mathbf{z}_t) = \frac{W_t^{1-\lambda}}{1-\lambda}\psi(t, \mathbf{z}_t), \tag{2.40}$$

which can be substituted in eq. (2.39) to yield optimal portfolio weights, and then in Bellman's equation (2.38) to yield a partial differential equation

involving only $\psi(\cdot,\cdot)$,

$$\psi_1 + (1-\lambda)(\mathbf{w}_t^{*\prime}\boldsymbol{\mu}_t^{\mathbf{P}} + r_f)\psi + \boldsymbol{\mu}_t^{\mathbf{z}\prime}\psi_2 - \frac{\lambda(1-\lambda)}{2}\mathbf{w}_t^{*\prime}\boldsymbol{\Sigma}_t^{\mathbf{P}}\mathbf{w}_t^{*}\psi + (1-\lambda)\mathbf{w}_t^{*\prime}\mathbf{D}_t^{\mathbf{P}}\boldsymbol{\rho}_t'\mathbf{D}_t^{\mathbf{z}\prime}\psi_2 + \frac{1}{2}\mathrm{tr}\big[\boldsymbol{\Sigma}_t^{\mathbf{z}}\psi_{2,2}\big] \;=\; 0, \quad (2.41)$$

with boundary condition $\psi(T,\cdot) = 1$.

Merton (1971) derived explicit solutions for the Hyperbolic Absolute Risk Aversion (HARA) family of utility functions (see §2.2.6/p. 56), which encompass the power utility case. In particular he showed that, for log-normally distributed assets and solving the case of the more general optimal consumption and investment problem,* that optimal consumption and investment policies have a form linear in current wealth,

$$C_t^* = a(t)W_t + b(t), \qquad \mathbf{w}_t^* W_t = g(t)W_t + h(t),$$

with a, b, g, h at most functions of time, if *and only if* the investor's utility functions on both consumption and terminal wealth belongs to the HARA family.

In more recent work, Kim and Omberg (1996) derive closed-form solutions for a number of specific parametrizations of the HARA utility in the case of a constant risk-free rate and a single risky asset with a stochastic risk premium following an Ornstein–Uhlenbeck (mean-reverting) process.†

Logarithmic Utility

Just as for the single-period problem, the logarithmic utility brings useful simplification in the multiperiod case. Logarithmic utility is a limiting form of the power utility where $\lambda = 1$. Substituting in eq. (2.41) yields the simplified equation

$$\psi_1 + \boldsymbol{\mu}_t^{\mathbf{z}\prime}\psi_2 + \frac{1}{2}\mathrm{tr}\big[\boldsymbol{\Sigma}_t^{\mathbf{z}}\psi_{2,2}\big] \;=\; 0,$$

subject to the boundary condition $\psi(T,\cdot) = 1$. It is obvious that the constant function $\psi(\cdot,\cdot) \equiv 1$ is a solution. Substituting in eq. (2.39), we see that the hedging demands term disappears (since $V_{2,3} \equiv 0$), leaving as the optimal solution only the myopic portfolio

$$\mathbf{w}_t^* = \big(\boldsymbol{\Sigma}_t^{\mathbf{P}}\big)^{-1}\boldsymbol{\mu}_t^{\mathbf{P}}.$$

A similar result also obtains in the discrete-time case. We now review the conditions under which the optimal multiperiod choice is in fact the myopic portfolio.

*See §2.2.6/p. 55.

†The risk premium is the excess return (over the risk-free rate) paid by the market for enticing investors to hold a risky asset.

When is the Myopic Policy Optimal?

Mossin (1968) examines the conditions for which the *myopic* portfolio choice is optimal.* From the results of eq. (2.30), eq. (2.39) and the previous section, we can summarize the conditions for optimality of the myopic policy as follows:

- The investment opportunities are fixed (for example, in the case of IID returns).
- The investment opportunities vary with time, but are unhedgeable; in this case, the $\boldsymbol{\rho}_t$ correlation matrix between state variables and asset returns is zero in eq. (2.39), and the induced hedging demands are also zero.
- The investor has a logarithmic utility on terminal wealth, as shown in the previous subsection.

2.2.4 The Martingale Formulation

The martingale formulation for optimal portfolios was introduced by Pliska (1986), Karatzas, Lehoczky, and Shreve (1987) and Cox and Huang (1989, 1991), and relies on a methodology established by Harrison and Kreps (1979) in the context of contingent claim valuation. Quite remarkably, this approach admits nearly the same class of problems as the optimal control formulation of §2.2.2/p. 44, yet dispenses with the arduous nonlinear Bellman partial differential equation (2.38), requiring solution only to a static optimization problem (and auxiliary subproblems, all much easier than eq. (2.38)). In a sense, it transforms an optimal control problem into a far simpler constrained optimization problem.

The approach is reviewed in some detail by Merton (1990, Chapter 6) and contrasted to the dynamic programming formulation. Because of its astuteness, we take time to outline the main ideas of the method, drawing from Merton's presentation but focusing on maximizing the utility of terminal wealth, omitting intermediate consumption.

The Growth-Optimum Portfolio

Let W_t be the value at time t of a portfolio that reinvests all earnings. Let $\mathrm{ACCR}(t,T)$ be the *average continuously compounded return* of the portfolio between times t and T,

$$\mathrm{ACCR}(t,T) \triangleq \frac{1}{T-t}\log\Big[\frac{W_T}{W_t}\Big].$$

*Recall that the myopic choice at time t depends only on the investment opportunity set and investor wealth at that time, disregarding future opportunities completely; in discrete time, it is equivalent to optimizing over the last period in the horizon.

Consider the trading strategy that maximizes this quantity; the resulting portfolio is called a *growth-optimum* portfolio. It is easy to see that it arises from having a log-utility on terminal wealth, $U(W_T) = \log W_T$, since

$$\begin{aligned}\mathbb{E}_t\big[\mathrm{ACCR}(t,T)\big] &= \frac{\mathbb{E}_t\big[\log W_T - \log W_t\big]}{T-t} \\ &\propto \mathbb{E}_t\big[\log W_T\big] - \log W_t\end{aligned}$$

where W_t is known at time t and of no impact in the optimal strategy. From the results of §2.2.3/p. 47, we have that the optimal portfolio weights in the risky assets, $\mathbf{w}_t^g$, can be written as

$$\mathbf{w}_t^g = (\boldsymbol{\Sigma}_t^{\mathbf{P}})^{-1}\boldsymbol{\mu}_t^{\mathbf{P}} \tag{2.42}$$

where $\boldsymbol{\Sigma}_t^{\mathbf{P}}$ is the instantaneous covariance matrix between asset returns at time t and $\boldsymbol{\mu}_t^{\mathbf{P}}$ is the vector of instantaneous expected excess asset returns. The fraction allocated to the risk-free asset is $1-\sum_{i=1}^N(\mathbf{w}_t^g)_i$.* As established previously, the growth-optimal portfolio rule is myopic.

Let X_t be the value of the growth-optimum portfolio at time t. From the posited asset-price dynamics of eq. (2.34), the dynamics of X_t are given as

$$\begin{aligned}\mathrm{d}X_t &= \left[\mathbf{w}_t^{g\prime}\Big[\frac{\mathrm{d}\mathbf{P}_t}{\mathbf{P}_t} - r_f\,\mathrm{d}t\Big] + r_f\,\mathrm{d}t\right]X_t \\ &= (\bar{\mu}^2 + r_f)X_t\,\mathrm{d}t + \bar{\mu}X_t\,\mathrm{d}z,\end{aligned} \tag{2.43}$$

where $\bar{\mu}^2 \triangleq \boldsymbol{\mu}_t^{\mathbf{P}\prime}(\boldsymbol{\Sigma}_t^{\mathbf{P}})^{-1}\boldsymbol{\mu}_t^{\mathbf{P}}$ and $\mathrm{d}z \triangleq \boldsymbol{\mu}_t^{\mathbf{P}\prime}(\boldsymbol{\Sigma}_t^{\mathbf{P}})^{-1}\mathbf{D}_t^{\mathbf{P}}\mathrm{d}\mathbf{B}_t^{\mathbf{P}}/\bar{\mu}$. The increment $\mathrm{d}z$ is a standard Wiener process.

The Cox-Huang Method

In continuous-time, if there is no intervening consumption, the optimal portfolio choice problem can be formulated as

$$\max_{\{\mathbf{w}_t\}_{t=0}^T}\ \mathbb{E}_0\big[U(W_T)\big], \tag{2.44}$$

subject to the budget constraint (2.36) and feasibility restriction $W_t \geq 0$ for all $t \leq T$. The dynamic programming solution studied previously expresses the optimal solution in a "feedback control" form, wherein the action depends on the current state of the process being controlled, $\mathbf{w}_t^* = \mathbf{w}^*(t, W_t, \mathbf{P}_t)$ (omitting other state variables $\mathbf{z}_t$ for simplicity). The expectation at time 0 is taken with respect to the joint distribution of asset prices $\mathbf{P}_t$ and current wealth dynamics W_t.

*As always, a negative fraction corresponds to borrowing at the risk-free rate.

Now consider a different problem specified as

$$\max_{\{\mathbf{w}_t\}_{t=0}^{T}} \tilde{\mathbb{E}}_0\big[U(\hat{W}_T)\big] \tag{2.45}$$

$$\text{subject to} \quad \hat{W}_T \geq 0,$$

$$X_0 \tilde{\mathbb{E}}_0\Big[\frac{\hat{W}_T}{X_T}\Big] \leq W_0, \tag{2.46}$$

where X_t is the value of the growth-optimum portfolio at time t, as defined previously. The expectation $\tilde{\mathbb{E}}$ is taken with respect to the joint distribution of asset prices $\mathbf{P}_t$ and values of the growth-optimum portfolio X_t. In particular, it does not include consideration of current wealth or other aspects of the controlled process. This implies that the optimal terminal wealth function $\hat{W}_T^* \equiv H(T, \hat{X}_T, \mathbf{P}_T; X_0, W_0, \mathbf{P}_0)$ *will not* have a "feedback control" form, since portfolio choices $\{\mathbf{w}_t\}_{t=0}^{T}$ and thence the wealth trajectory $\hat{W}_t$ have no bearing on the asset price process $\mathbf{P}_t$ or value of the growth-optimum portfolio X_t.* Put differently, eq. (2.45) can be solved by using the Karush-Kuhn-Tucker (KKT) conditions for *static* constrained optimization.

The connection between problems (2.44) and (2.45) is established by the following result by Cox and Huang (1991):

Theorem 4 (Cox–Huang Equivalence) *Under quite mild regularity conditions, there exists a solution to* (2.44) *if and only if (a) there exists a solution to* (2.45)*, and (b)* $W_T = \hat{W}_T^*$.

In-depth economic intuition behind this result is provided by Merton (1990, Chapter 16). In the remainder of this section, we derive the optimal portfolio rule $\mathbf{w}_t^*$ that arises from solution to eq. (2.45). We start by incorporating the constraints into the objective by way of Lagrange multipliers,

$$\max \tilde{\mathbb{E}}_0\left[U(\hat{W}_T) + \lambda_1\Big[W_0 - \frac{X_0 \hat{W}_T}{X_T}\Big] + \lambda_2 \hat{W}_T\right], \tag{2.47}$$

where $\lambda_1, \lambda_2 \geq 0$. For all X_t and $\mathbf{P}_t$ (having positive probability in the above expectation), the first-order condition for optimality is written as

$$U_1(\hat{W}_T) = \lambda_1 \frac{X_0}{X_T} - \lambda_2. \tag{2.48}$$

The substantive analysis can proceed assuming that constraint (2.46) is binding with equality, for otherwise the investor's initial wealth is sufficient to ensure satiation given his utility function, and the optimal policy is therefore to invest in the riskless asset.† The assumption of non-satiation ensures that for any terminal wealth W_T, we have strictly positive marginal utility $U_1(W_T) > 0$ and strict concavity of utility, $U_{1,1}(W_T) < 0$. Non-satiation

*Assuming negligible market impact.

†See Merton (1990, p. 174) for a detailed argument.

in turn implies that the shadow price of wealth, λ_1, is strictly positive in eq. (2.47).

From the KKT condition* $\lambda_2 \hat{W}_T^* = 0$ and eq. (2.48), we have

$$\lambda_2 = \max\Big[0, \lambda_1 \frac{X_0}{X_T} - U_1(0)\Big]. \tag{2.49}$$

Furthermore, since $U_{1,1}(\cdot) < 0$, U_1 is invertible. Let $R(y) \triangleq U_1^{-1}(y)$. From eq. (2.48) and (2.49), we determine the optimal terminal wealth to be

$$\begin{aligned}
\hat{W}_T^* &= R\bigg[\lambda_1 \frac{X_0}{X_T} - \max\Big[0, \lambda_1 \frac{X_0}{X_T} - U_1(0)\Big]\bigg] \\
&= R\bigg[\max\Big[\lambda_1 \frac{X_0}{X_T}, U_1(0)\Big]\bigg] \\
&= \max\bigg[R\Big[\lambda_1 \frac{X_0}{X_T}\Big], R\big(U_1(0)\big)\bigg] \\
&= \max\bigg[R\Big[\lambda_1 \frac{X_0}{X_T}\Big], 0\bigg],
\end{aligned} \tag{2.50}$$

taking advantage of the monotonicity of $R(\cdot)$ to exchange R and max in the third step. The solution to $\hat{W}_T^*$ only requires the determination of λ_1. As indicated above, this is possible assuming that constraint (2.46) is binding with equality. Under this condition, substituting eq. (2.50) in (2.46) yields the following transcendental algebraic equation

$$\tilde{\mathbb{E}}_0\bigg[\frac{\max\Big[R\big[\lambda_1 \frac{X_0}{X_T}\big], 0\Big]}{X_T}\bigg] - \frac{W_0}{X_0} = 0$$

whose solution for λ_1 depends only on the initial conditions and can be expressed as $\lambda_1 = \lambda_1(X_0, \mathbf{P}_0, W_0)$. For compactness, we shall express $\hat{W}_T^*$ as

$$\hat{W}_T^* \equiv H(T, X_T, \mathbf{P}_T), \tag{2.51}$$

noting that the implicit functional dependence of H on X_0, $\mathbf{P}_0$, and W_0 is omitted for simplicity. We can derive the optimal portfolio strategy $\mathbf{w}_t^*$ as follows. Define the function $F(t, X_t, \mathbf{P}_t)$ as

$$F(t, X_t, \mathbf{P}_t) \triangleq X_t \tilde{\mathbb{E}}_t\bigg[\frac{H(T, X_T, \mathbf{P}_T)}{X_T} \,\bigg|\, X_t, \mathbf{P}_t\bigg].$$

From (2.46), we have that $F(0, X_0, \mathbf{P}_0) = W_0$. At an arbitrary time t, assuming the investor acts optimally since time 0, he faces from time t the

*See, e.g., Luenberger and Ye (2007) for more details on the KKT conditions for constrained optimization.

same problem (2.45)–(2.46) faced from time 0,

$$\max_{\{\mathbf{w}_\tau\}_{\tau=t}^{T}} \tilde{\mathbb{E}}_0\big[U(\hat{W}_T)\big] \tag{2.52}$$

$$\text{subject to} \quad \hat{W}_T \geq 0,$$

$$X_t \tilde{\mathbb{E}}_t\Big[\frac{\tilde{W}_T}{X_T}\Big] \leq W_t. \tag{2.53}$$

Because eq. (2.51) is an intertemporal optimum, it must be the case that H is also the rule the investor would follow as a solution to eq. (2.52) and assuming no satiation, constraint (2.53) remains satisfied with equality. By the definition of F and (2.53), we have

$$W_t = F(t, X_t, \mathbf{P}_t). \tag{2.54}$$

From Itô's lemma, asset-price dynamics (2.34) and the dynamics of the growth-optimal portfolio X_t (2.43), we have, after some algebra, that the optimal wealth dynamics are given by

$$\mathrm{d}F = \bar{\alpha}F\,\mathrm{d}t + F_2\bar{\mu}X_t\,\mathrm{d}z + (F_3 \odot \mathbf{P}_t)'\mathbf{D}_t^{\mathbf{P}}\mathrm{d}\mathbf{B}_t^{\mathbf{P}} \tag{2.55}$$

with

$$\bar{\alpha}F \equiv F_1 + (\bar{\mu}^2 + r_f)XF_2 + \frac{1}{2}\bar{\mu}^2X^2F_{2,2} + (F_3 \odot \mathbf{P}_t)'(\boldsymbol{\mu}_t^{\mathbf{P}} + r_f) + \frac{1}{2}\mathrm{Tr}\big[F_{3,3}\boldsymbol{\Sigma}_t^{\mathbf{P}}\big] + X(F_{2,3} \odot \mathbf{P}_t)'\boldsymbol{\mu}_t^{\mathbf{P}}$$

where the $\odot$ operator signifies element-wise multiplication of vector elements, i.e. $(\mathbf{x} \odot \mathbf{y})_i \stackrel{\triangle}{=} \mathbf{x}_i\,\mathbf{y}_i$.

Theorem 5 (Cox–Huang Optimal Weights) *If there exists an optimal solution to problem (2.45), $\hat{W}_T^*$, then for $t \leq T$ the optimal portfolio strategy $\{\mathbf{w}_t^*\}$ that achieves this allocation is given by*

$$\mathbf{w}_t^* W_t = F_2(t, X_t, \mathbf{P}_t)\,X_t\,\mathbf{w}_t^g + F_3(t, X_t, \mathbf{P}_t) \odot \mathbf{P}_t \tag{2.56}$$

with the balance of the investor's wealth, $1 - \boldsymbol{\iota}'\mathbf{w}_t^*$, in the riskless asset, where $\mathbf{w}_t^g$ is given by eq. (2.42).

Proof Let $\{\mathbf{w}_t^*\}$ denote the optimal allocations in the risky assets. From eq. (2.36), the dynamics of wealth under optimal allocation are given as

$$\mathrm{d}W = \big(\mathbf{w}_t^{*\prime}\boldsymbol{\mu}_t^{\mathbf{P}} + r_f\big)W_t\,\mathrm{d}t + W_t\mathbf{w}_t^{*\prime}\mathbf{D}_t^{\mathbf{P}}\mathrm{d}\mathbf{B}_t^{\mathbf{P}}. \tag{2.57}$$

But from eq. (2.54), we must have $\mathrm{d}W - \mathrm{d}F \equiv 0$ for all $t \leq T$, and comparing eq. (2.55) and (2.57) this can be satisfied if and only if

$$\bar{\alpha}F = \big(\mathbf{w}_t^{*\prime}\boldsymbol{\mu}_t^{\mathbf{P}} + r_f\big)W_t \tag{i}$$

and

$$F_2\bar{\mu}X_t\,\mathrm{d}z + (F_3 \odot \mathbf{P}_t)'\mathbf{D}_t^{\mathbf{P}}\mathrm{d}\mathbf{B}_t^{\mathbf{P}} = W_t\mathbf{w}_t^{*\prime}\mathbf{D}_t^{\mathbf{P}}\mathrm{d}\mathbf{B}_t^{\mathbf{P}}. \qquad \text{(ii)}$$

Replacing $\mathrm{d}z$ by its definition, we simplify the common terms $\mathbf{D}_t^{\mathbf{P}}\mathrm{d}\mathbf{B}_t^{\mathbf{P}}$ on both sides of (ii) and obtain the result. ■

By virtue of Theorem 4, the optimal weights found from eq. (2.56) are also those that solve the original problem (2.44).

The only remaining hurdle in applying the method is to obtain the distribution $P(X_t, \mathbf{P}_t \mid X_0, \mathbf{P}_0), 0 \le t \le T$, under which the expectation $\tilde{\mathbb{E}}$ can be evaluated. This is possible by solving a backward Kolmogorov equation (Merton 1990; Wilmott 2006), a linear parabolic partial differential equation, itself much easier to solve than the nonlinear Bellman equation.

Cox and Huang (1989) derive explicit solutions, in the presence of nonnegativity constraints on consumption and final wealth, for hyperbolic absolute risk aversion (HARA) utility functions when the asset prices follow a geometric Brownian motion. The nonnegativity constraints cause the optimal policies to no longer be linear in the moments of the return distribution. Wachter (2002) finds closed-form solutions for mean-reverting returns when markets are assumed to be complete.

Implementation

A number of implementations of this method have been presented in the literature. Cvitanić, Goukasian, and Zapatero (2003) introduce a relatively simple method based on pure Monte Carlo simulation for approximating the expectations required for computing optimal portfolios; their method assumes that asset-price and state variable dynamics are known. A different simulation approach, by Detemple, Garcia, and Rindisbacher (2003), derives explicit components for the hedging demand terms of optimal portfolios using the Malliavin calculus and generalizes earlier results by Ocone and Karatzas (1991); the approach allows a large number of assets and state variables, assumed to follow a diffusion process, to be used. The convergence and efficiency properties of the Malliavin derivatives, in contrast to PDE and other Monte Carlo estimators, are analyzed. In more recent work, Aït-Sahalia and Brandt (2007) use option-market prices to directly infer state prices; they find significantly different optimal consumption and investment policies than those arising from standard assumptions on asset return dynamics.

Of a related flavor is the work by Brandt and Santa-Clara (2006) who consider augmenting the asset space by a set of managed portfolios, both *conditional portfolios* that are proportional to conditioning variables, and *timing portfolios* that invest in one asset for one time period (at some point in the future) and do not invest during other periods. These assets are similar in spirit to the Arrow–Debreu securities which form the theoretical foundation of the Cox–Huang method. Brandt and Santa-Clara show that solving a *static* Markowitz mean-variance problem on the augmented asset space can

quite well approximate a dynamic strategy for medium-term horizons (up to five years), despite being much simpler to implement.

2.2.5 Investor Learning

In the portfolio choice literature, "learning" generally refers to the investor's gradually-better modeling of the generating distribution of asset returns, which may be conditional or not. In general, the optimal decision depends on the fact that we expect to learn about future changes in expected returns, which induces a negative hedging demand in the risky asset.* Kandel and Stambaugh (1996) and Barberis (2000) examine how asset return predictability and parameter estimation uncertainty affect the optimal allocations; both are found to induce sizable horizon effects, and bring substantial allocation differences that are exacerbated at long horizons. Xia (2001) discusses the effects of parameter uncertainty in a multiperiod context; it is found that the opportunity cost of ignoring predictability or learning is quite substantial. Brandt, Goyal, Santa-Clara, and Stroud (2005) propose a simulation approach to solve discrete-time dynamic portfolio choice problems involving non-standard preferences, a large number of assets and a large number of state variables, based on the well-known Longstaff and Schwartz (2001) approximation method originally proposed in the context of financial derivatives.

Finally, Skoulakis (2007) considers a fully Bayesian investor operating in discrete time and that solves a portfolio choice problem while simultaneously updating beliefs about the parameters of the generating distribution, considering that returns may (partially) be predictable. He finds that in the presence of predictability, learning reduces, without completely eliminating, the positive hedging demands that are normally induced by predictability.

2.2.6 Common Extensions

Beyond the basic multiperiod framework of Samuelson and Merton, a large number of extensions have been proposed to address the shortcomings of the original formulations. In addition to the classical issues of consumption and labor income, extensive work has been pursued in the areas of non-standard preferences and utility functions, and characterization of the optimal policy in the presence of transaction costs, taxes and other frictions.

Intermediate Consumption and Labor Income

Intermediate consumption has traditionally been part of the multiperiod optimal investment problem since Samuelson (1969) and Merton (1969). In these settings, the problem is formulated so as to assume a single consumption good and postulates a time-separable utility over consumption.

*Put differently, this means that we desire less of the asset today, given that we expect to know more about its distribution with more observations in the future.

In continuous time, the investor's objective is then to jointly maximize the utility of the consumption path and terminal wealth,

$$\max_{\{C_t,\mathbf{w}_t\}_{t=0}^{T}} \mathbb{E}_0\Big[\int_0^T U_C(C_t,t)\,\mathrm{d}t + U_T(W_T)\Big],$$

where $U_C(\cdot,\cdot)$ is the utility of the consumption rate C_t at time t, and U_T is the utility of the terminal wealth, subject to a modified budget constraint that accounts for consumption,

$$\mathrm{d}W_t = W_t\big(\mathbf{w}_t'\boldsymbol{\mu}_t^{\mathbf{P}} + r_f\big)\mathrm{d}t - C_t\mathrm{d}t + W_t\mathbf{w}_t'\mathbf{D}_t^{\mathbf{P}}\mathrm{d}\mathbf{B}_t^{\mathbf{P}}.$$

Nonstochastic labor income is just as easily incorporated by adding it into the budget constraint, as was shown in Merton (1971). The problem of stochastic labor income was treated by Koo (1998) and Viceira (2001) among others.

Non-Standard Preferences

Merton (1971) derives explicit solutions for the consumption–investment problem when investors have a time-separable utility over a consumption C that can be expressed as

$$U(C,t) = \exp(-\rho\,t)\,V(C),$$

with ρ a discount factor and V a utility function whose absolute risk aversion is positive and hyperbolic in its argument (i.e. belonging to the *hyperbolic absolute risk aversion*, or HARA, family),

$$V(C) = \frac{1-\gamma}{\gamma}\Big(\frac{\beta C}{1-\gamma} + \eta\Big)^{\gamma}, \tag{2.58}$$

subject to

$$\gamma \neq 1, \qquad \beta > 0, \qquad \frac{\beta C}{1-\gamma} + \eta > 0, \qquad \eta = 1 \text{ if } \gamma = -\infty.$$

With suitable choice of parameters, the HARA family encompasses both Constant Absolute Risk Aversion (CARA) and Constant Relative Risk Aversion (CRRA) utilities.

As discussed in §2.1.4/p. 14, CRRA preferences are the only ones for which asset proportions are independent of wealth, and this—along with its analytical tractability—make it a popular choice in the literature. However, it can be shown (e.g. Campbell and Viceira 2002) that utility functions of this class intrinsically link the risk aversion with what is known as the *elasticity of intertemporal substitution* (the propensity to substitute consumption between periods). For this reason, Epstein and Zin (1989) introduced a class of recursive utility functions that generalize the CRRA class and admit independent risk aversion and coefficient of intertemporal substitution. Campbell

and Viceira (1999) and Schroder and Skiadas (1999) analyze portfolio and consumption choices under this more general class of utility functions.

In a different vein, there has been in recent years an explosion of studies in the broad area of *behavioral finance*, where market participants are not assumed to always make rational choices.* In an asset allocation context, Shefrin and Statman (2000) apply the *prospect theory* of Kahneman and Tversky (1979) to construct a behavioral portfolio theory (BPT) and show that, in general, the "behavioral" efficient frontier does not coincide with the Markowitz one. In particular, BPT investors are simultaneously risk averse and risk seeking, and construct portfolios that consist of both bonds and lottery tickets. In recent work, Vlcek (2006) finds that in a two-period setting, an investor governed by prospect theory is not prone to the disposition effect,† his behavior instead essentially being driven by loss aversion: first-period gains cushion possible future losses and encourage increased risk-taking in the second period. Berkelaar, Kouwenberg, and Post (2004) extend the martingale formulation of §2.2.4/p. 49 to analyze the optimal investment strategy of loss-averse investors. Brandt (2004) provides a broader survey of this literature.

Transaction Costs

In continuous time, the absence of transaction costs causes the investor to continuously rebalance his portfolio, inducing an unrealistic level of trading activity. The effect of transaction costs in the continuous-time framework has long been studied, starting with Magill and Constantinides (1976) in the context of assets following a geometric Brownian motion with a HARA utility function in the presence of *proportional costs* (*cf.* eq. 2.12). They find that the optimal policy is characterized by an "envelope" around the time-t optimal portfolio weights (the targets). The optimal policy is not to trade when the current portfolio holdings are contained within the envelope, but to rebalance *up to the envelope* (but not up to the target) when they fall outside. This induces *random*, rather than continuous, portfolio rebalancing. This policy is found identical in functional form to the classical Bellman, Glicksberg, and Gross (1955) ordering policy for the infinite-horizon multi-commodity inventory problem with proportional ordering costs. The intuitive justification for the presence of this envelope is that when existing holdings are close the the optimal (the targets at time t), there are only second-order utility gains to be made from adjusting the portfolio, but first-order transaction costs to bear.

Taksar, Klass, and Assaf (1988) and Davis and Norman (1990) also studied related problems in the one-asset case, the latter relating it to the solution

*For comprehensive reviews of this vast field, see, e.g. Shefrin (2002) and Montier (2002); in an asset pricing context, Shefrin (2005) provides an in-depth treatment.

†The *disposition effect* refers to the empirical tendency of investors to prematurely sell winners and hold onto losers (Shefrin and Statman 1985).

of a nonlinear free boundary problem. Cvitanić and Karatzas (1996) introduced a solution based on the martingale approach. Shreve and Soner (1994) analyzed the problem in terms of the viscosity solutions to the Hamilton-Jacobi-Bellman (HJB) equation. Leland (2000) generalizes the study to the multi-asset case (and also simultaneously considers capital-gain taxes), in the context of a *portfolio implementation* problem faced by a practitioner in which the target weights are provided exogenously. He characterizes an approximate no-trade region in terms of the 2^N corner points of the boundary surface. For a long time, it remained a practical problem that the runtime of the best existing solution methods to the HJB equation would grow super-exponentially with dimension N (the number of assets in the portfolio), making them impractical for portfolios of more than about $N = 4$ assets. Recently, Muthuraman and Zha (2008) proposed a simulation-based approach to tackle this problem which scales polynomially in dimension, while providing close fits to existing solutions.

The problem of *fixed transaction costs* was addressed by Eastham and Hastings (1988) in an optimal consumption–investment context; they derive a solution through quasi-variational inequalities of the value function. A numerical solution method was proposed by Atkinson, Pliska, and Wilmott (1997) that scales well to moderate-sized portfolios (30 assets). Liu (2004) considered the case of a constant absolute risk aversion (CARA) investor dealing with both fixed and proportional transaction costs; his analysis reveals that costs can reduce the significance of asset return predictability on optimal portfolio rules.

Finally Morton and Pliska (1995) considered trading costs that are proportional to a fixed fraction of portfolio value, purely as a means to discourage frequent trading; they relate the solution to that of a stopping time problem. However, this "proportional-to-wealth" cost structure seems quite disconnected from that faced by real investors.

Taxes and Other Frictions

Capital gain taxes add another dimension to the transaction-cost issues. In many jurisdictions, including the United States, capital gains made when an asset is sold are taxable, although the tax rate may depend on the holding period of the asset (making short-term holdings more subject to penalty), and have a basis for calculating the tax amount that depends on the price at which the asset was originally bought (the tax basis). Furthermore, assets sold at a loss may offset gains from other assets. In contrast to simple handling of transaction costs, which depend only on local information, capital gain taxes complicate the solution of the backward dynamic programming equations (2.26) in the multiperiod case, since the buying price of an asset is unknown when "solving back in time". This can be overcome, approximately, by increasing the state space (recording, for each asset, not only the amount held in the portfolio, but also the original buy-date and

buy-price), although this approach is clearly limited by the curse of dimensionality; moreover, an exact solution formulation grows exponentially with the number of time periods.* For single-period problems, Elton and Gruber (1978) first considered the question of capital gain taxes. In a multiperiod setting, Dammon, Spatt, and Zhang (2001, 2004) and Gallmeyer, Kaniel, and Tompaidis (2006) approximate the tax basis by the weighted average purchase price. DeMiguel and Uppal (2005) show how to use the exact tax basis using a nonlinear programming formulation, and report that the certainty-equivalent loss from using an approximate basis is small. More recently, Osorio, Gülpinar, and Rustem (2008) apply a multistage stochastic programming approach (§2.3.6/p. 65) to this problem.

Cvitanić (2001) reviews the substantial literature that applies the Martingale formulation (§2.2.4/p. 49) to problems with frictions.

2.3 Direct and Alternative Methods for Portfolio Choice

In the spirit of the original Markowitz methodology, all the portfolio allocation methods covered up to this point follow a functional separation that can be summarized as

1. Estimate the (conditional) distribution (or moments thereof) of asset returns, from historical data and conditioning variables;

2. Construct an optimal portfolio (policy) by maximizing a utility function.

However, from a complete-system viewpoint, nothing prevents a more direct link between conditioning variables and allocation decisions to be made. This can be motivated from several perspectives. First, we already discussed (§2.1.8/p. 31) the impact of estimation error on the stability of the resulting allocations, as well as the complex arsenal of methods that have been proposed to remedy one aspect or another of the problem. It can be shown that estimation errors compound in a multiperiod setting, making a bad problem worse (Brandt 2004). Second, arguments from statistical learning theory (Vapnik 1998) can be made to the effect that to solve problem X, given a limited amount of data (here, historical realizations of financial series), one should not first attempt to solve a harder problem Y. In the context of portfolio allocation, the really hard problem is the high-dimensional estimation of the conditional distribution of asset returns (which involves at least $O(N^k)$ quantities, where N is the number of assets, and k is the number of moments in the distribution that we wish to represent), whereas the asset

*Which requires recording the buy-date and buy-price for every transaction.

allocation itself involves only $O(N)$ quantities—the weight to be given to each asset.

The idea of directly obtaining portfolio weights from explanatory variables has first been explored in the machine learning community and has more recently been studied in the financial economics literature as well. We also review "non-allocation" approaches—mostly based on reinforcement learning—that do not attempt to produce a genuine allocation of capital among assets but rather output a "long–short" decision aimed at short-term trading. Finally we conclude with an overview of approaches based on stochastic programming, studied mostly in operations research.

2.3.1 Machine Learning Approaches

The first applications of supervised learning algorithms to financial decision-making, and portfolio construction, problems have mainly focused on non-linear approaches to forecasting (e.g. Weigend and Gershenfeld 1993).* It goes without saying that these are simply generalizations of linear factor models (§2.1.7/p. 25) and do not sidestep the intrinsically difficult task of estimating the conditional distribution of asset returns.

Starting in the mid-1990's several asset-allocation approaches based on direct maximization of financial criteria started to appear. Choey and Weigend (1997) used a feedforward neural network (Rumelhart, Hinton, and Williams 1986) trained to directly maximize a Sharpe Ratio criterion to make an allocation decision between one risky (the DAX index of German stocks) and one riskless asset. Bengio (1997) trains a neural network on a profit criterion that accounts for transaction costs using a differentiable representation in the objective function, and compares against a network trained to optimize a forecasting criterion (the mean-squared error) on a basket of Canadian stocks; he reports significantly better out-of-sample risk-adjusted trading performance in favor of the financial criterion. In related work, Ghosn and Bengio (1997) analyze the parameter-sharing ability of multi-task learning to help improve forecasting performance across a universe of stocks, where each stock is viewed as a single task.

Chapados (2000) (see also Chapados and Bengio 2001) applied recurrent neural networks to a mean-VaR framework (*cf.* §2.1.5/p. 17), showing how the network can be trained to directly maximize expected return while satisfying a target portfolio risk constraint and minimize transaction costs. He compared against standard benchmarks including mean–variance optimization (where expected returns are forecast with a feedforward neural network and the covariance matrix is obtained by a standard RiskMetrics (1996) estimator) and obtains statistically significant out-of-sample financial performance in excess of the benchmark index when allocating to the 14 subsectors of the Canadian TSE-300 index.

*More recent work include books by Shadbolt and Taylor (2002), Dunis, Laws, and Naïm (2003), and McNelis (2005).

Dunis, Laws, and Evans (2006a, 2006b) report good out-of-sample performance results using recurrent neural networks applied to trading commodity spread portfolios, beating standard feedforward neural networks and other benchmarks.

Zimmerman and colleagues have worked for a number of years on multitiered recurrent neural architectures that attempt to capture specific dynamical features of financial time series. Zimmermann, Neuneier, and Grothmann (2001) apply a non-linear generalization of an ARMA$(1,1)$ model, termed an error-correcting neural network (ECNN), to forecast a set of price series. These forecasts are then subject to a second-level parameterized allocation function that determines an allocation $\mathbf{w}_{t,i}$ from a "softmax" transformation on asset-weighted excess returns. Let $\mathbf{f}_{t,i}$ be the expected-return forecast produced at time t for asset i by the ECNN; the portfolio weight is given by the second-level allocation function as

$$\mathbf{w}_{t,i} = \frac{\exp \mathbf{e}_{t,i}}{\sum_{j=1}^{N} \exp \mathbf{e}_{t,j}},$$

where $\mathbf{e}_{t,i}$ is the weighted average "excess" return of asset i against all other assets

$$\mathbf{e}_{t,i} = \sum_{j=1}^{N} \beta_{i,j}(\mathbf{f}_{t,i} - \mathbf{f}_{t,j}).$$

The parameters $\beta_{i,j}$ are optimized to maximize the in-sample total return subject to a deviation constraint from a benchmark. The authors claim that assets giving unreliable forecasts have their associated β coefficients pushed to zero, thereby implicitly controlling portfolio risk. They report risk-adjusted excess return on a stock–bond allocation task among the G7 countries with respect to an (unspecified) benchmark. More recently, the same group generalized these networks to operate at multiple time scales and applied them to the forecasting of foreign exchange (Zimmermann, Grothmann, Schäfer, and Tietz 2006; Zimmermann, Bertolini, Grothmann, Schäfer, and Tietz 2006).

2.3.2 Parametric Portfolio Policies

Brandt, Santa-Clara, and Valkanov (2007) introduce an approach where the portfolio weight given to a stock directly depends on the *specific features* that characterize a stock though a parameterized functional form, $w_{i,t} = f(\mathbf{x}_{i,t}; \boldsymbol{\theta})$. In particular, they consider linear policies of the form

$$w_{i,t} = \bar{w}_{i,t} + \frac{1}{N_t} \boldsymbol{\theta}' \hat{\mathbf{x}}_{i,t},$$

where $w_{i,t}$ is the weight of asset i at time t in the portfolio, $\bar{w}_{i,t}$ is a benchmark weight (e.g. $1/N$ or the weight in a capitalization-weighted market portfolio), $\boldsymbol{\theta}$ is a fixed vector of coefficients (to be estimated) and $\hat{\mathbf{x}}_{i,t}$ is a

vector of stock-dependent characteristics standardized cross-sectionally (at time t) to have a zero-mean and unit-standard deviation across all stocks. By construction (due to the standardization), the portfolio weights sum to one if the benchmark weights sum to one: the "correction term" $\frac{1}{N_t}\boldsymbol{\theta}'\hat{\mathbf{x}}_{i,t}$ can be interpreted as a direct specification of the *active risk* of the position in asset i. The coefficients are optimized to maximize the expected utility of the one-period portfolio returns

$$\max_{\boldsymbol{\theta}} \mathbb{E}_t\big[U(R_{P,t+1})\big] = \max_{\boldsymbol{\theta}} \mathbb{E}_t\left[U\Big(\sum_{i=1}^{N_t} f(\mathbf{x}_{i,t};\boldsymbol{\theta})\mathbf{R}_{i,t}\Big)\right],$$

where the expectation is evaluated empirically on past data. Note that the parameters $\boldsymbol{\theta}$ are fixed across both time and stocks. Note also that the approach applies effortlessly to a variable number of stocks during each period (indicated by the upper summation index N_t). Using only three conditioning variables* and all stocks from the CRSP–Compustat database from 1964 to 2002, this simple method generates statistically significant out-of-sample returns in excess of the benchmark, after transaction costs.

2.3.3 Nonparametric Portfolio Weights

Tying in more directly with the optimal multiperiod portfolio choice formulation, Brandt (1999) considers the sample analogues of the first-order optimality conditions† given by eq. (2.27). For single-period portfolio choice, this equation can be written

$$\mathbb{E}_t\big[U'(\mathbf{w}_t'\mathbf{R}_{t+1})\mathbf{R}_{t+1}\big] = 0.$$

The idea is to estimate this expectation by a sample analogue (historical data), use a nonparametric estimator (Härdle 1990) to weigh each observation according to how "close" it is to a given test state variable $\mathbf{z}$ and numerically solve for the portfolio weights $\mathbf{w}_t$ that satisfy the equation,

$$\hat{\mathbf{w}}_t(\mathbf{z}) = \left\{\mathbf{w} : \frac{1}{T}\sum_{t=1}^{T} k_{h_T}\big(\mathbf{z}_t - \mathbf{z}\big)U'(\mathbf{w}'\mathbf{R}_{t+1})\mathbf{R}_{t+1} = 0\right\},$$

where $k_{h_T}(\cdot)$ is a kernel function (which we assumed is normalized), h_T is a kernel bandwidth parameter. The approach can be generalized to the multiperiod case by backward induction, assuming a CRRA utility function.

*Consisting of (i) the log market equity, (ii) the log book-to-market ratio, and (iii) the lagged one-year return, defined as the compounded return between months $t-13$ and $t-2$. The first two variables are used six months after their nominal validity date to ensure an adequate delay for the diffusion of financial statement information. Some experiments also added the slope of US interest rates yield curve as a conditioning variable for the other three.

†Also called Euler equations.

Unfortunately, this approach suffers from the curse of dimensionality in the number of state variables. Aït-Sahalia and Brandt (2001) propose an approach wherein an optimal linear projection down to a single state variable is found before applying the above kernel regression. This can be used to perform variable selection at the level of the state variables.

2.3.4 "Non-Allocation" Approaches

We call "non-allocation" approaches those that do not aim at solving a full portfolio problem (including taking advantage of diversification across assets) but perhaps the simpler problem of deciding whether the investor should be "long" (buy) or "short" (sell) in an asset. Most of the following approaches are based on reinforcement learning, which is reviewed in §3.2/p. 77.*

Neuneier (1996) uses a Q-learning algorithm (Watkins and Dayan 1992) to learn the value function of a risk-neutral investor on the foreign-exchange and the stock market; Neuneier (1998) and Neuneier and Mihatsch (1999) generalize the approach to a multi-task setting and can derive multiperiod portfolio policies that account for transaction costs and risk-averse utility functions.

Ormoneit and Glynn (2001) introduce a non-parametric estimator of the value function for reinforcement learning and apply it to an allocation task between a risky and risk-free asset under logarithmic utility, where the decision is discretized (the fraction invested in the risky asset can be in the set $\{0, 0.1, 0.2, \ldots, 1.0\}$) and the state variable is proportional to the estimated risky asset volatility.

Moody, Wu, Liao, and Saffell (1998) and Moody and Saffell (2001a) introduce a "direct reinforcement" approach that sidesteps learning the value function and directly learns an allocation policy. They suggest an approximation scheme based on a Taylor series expansion to optimize a Sharpe Ratio criterion, which normally does not lend itself well to a reinforcement learning objective since it is not time-separable. More recently, Hens and Wöhrmann (2007) revisited the method in the context of strategic asset allocation between stocks and bonds for the US, UK, Germany, and Japan markets, for a power utility investor. The policy function is strictly determined by the forecasted yield spread between stocks and bonds (determined from average historical returns), which constitutes the only input variable. The learned policy suggests that this spread has significant explanatory power for market timing.

*Note that this overview must omit coverage of the vast fields of "automated trading systems" and "technical analysis". See, e.g. Kaufman (1998), for an introduction.

2.3.5 Information-Theoretic Approaches

Approaches based on information theory (Cover and Thomas 2006) have also been investigated, although not traditionally by the financial economics community. Cover (1991) introduced *universal portfolios* which guarantee asymptotic performance equal to the best (in hindsight) *constant* portfolio weights.* Consider a fixed portfolio $\mathbf{w}, \sum_{i=1}^{N} \mathbf{w}_i = 1$, and let $S_k(\mathbf{w})$ be the cumulative portfolio return over a fixed horizon $t = 1 \dots k$,

$$S_k(\mathbf{w}) \triangleq \prod_{t=1}^{k} \mathbf{w}'(1 + \mathbf{R}_t).$$

Let S_T^* be the maximum achievable wealth over a horizon-T given price sequence,

$$S_T^* = \max_{\mathbf{w} \in \mathcal{W}} S_T(\mathbf{w})$$

where $\mathcal{W}$ consists of the set of nonnegative-weight portfolios whose weights sum to one,

$$\mathcal{W} = \Big\{ \mathbf{w} \in \mathbb{R}^N : \mathbf{w}_i \geq 0, \sum_{i=1}^{N} \mathbf{w}_i = 1 \Big\}.$$

The UNIVERSAL portfolio strategy is simply defined as the performance-weighted constant portfolio average over a past history, namely

$$\hat{\mathbf{w}}_1 = \Big(\frac{1}{N}, \frac{1}{N}, \cdots, \frac{1}{N} \Big) \qquad \text{and} \qquad \hat{\mathbf{w}}_{k+1} = \frac{\int_{\mathcal{W}} \mathbf{w}\, S_k(\mathbf{w})\, \mathrm{d}\mathbf{w}}{\int_{\mathcal{W}} S_k(\mathbf{w})\, \mathrm{d}\mathbf{w}}.$$

Denote by $\hat{S}_T$ the wealth achieved by the UNIVERSAL portfolio strategy over the horizon T. Cover proved that for arbitrary bounded price sequences, the wealth achieved by the UNIVERSAL strategy grows as that of the best constant portfolio weights,†

$$(1/n) \ln \hat{S}_T - (1/n) \ln S_T^* \to 0.$$

Remarkably, this result does not depend on any statistical assumption on the behavior of the price sequences. Cover and Ordentlich (1996) considered the addition of side information (i.e. explanatory variables, albeit discrete ones) and obtains precise bounds on the ratio of the wealth given by the universal portfolio to the best wealth achievable by a constant rebalanced portfolio given hindsight. Ordentlich and Cover (1998) extended the results to an adversarial setting with bounds on achievable wealth in a game wherein a participant must announce a causal portfolio strategy at the outset and

*Note that the strategy of keeping constant portfolio weights implies continuous rebalancing of the portfolio to keep the actual portfolio weights equal to their (constant) targets: as prices change, so do portfolio weights, which implies the necessity of rebalancing.

†Which is only known in hindsight and therefore unachievable.

an opponent is allowed to choose any stock market sequence and the best constant rebalanced portfolio for that sequence.

Blum and Kalai (1998) addressed the original lack of consideration of transaction costs in Cover's formulation. In subsequent work, they also presented an efficient randomized approximation of the original algorithm that overcomes its exponential-time complexity (Kalai and Vempala 2002).

2.3.6 Stochastic Programming Approaches

Stochastic programming*(Dantzig 1955; Birge and Louveaux 1997) is a generalization of mathematical programming to optimization problems involving random variables. A basic formulation of the problem is the following *two-stage stochastic linear program with fixed recourse*,

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{c}'\mathbf{x} + \mathbb{E}_{\boldsymbol{\xi}}\big[Q(\mathbf{x}, \boldsymbol{\xi})\big] \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b}, \\ & \mathbf{x} \geq 0, \end{aligned} \tag{2.59}$$

where

$$\begin{aligned} Q(\mathbf{x}, \boldsymbol{\xi}) = \min_{\mathbf{y}} \quad & \mathbf{q}'\mathbf{y} \\ \text{subject to} \quad & \mathbf{W}\mathbf{y} = \mathbf{h} - \mathbf{T}\mathbf{x}, \\ & \mathbf{y} \geq 0 \end{aligned}$$

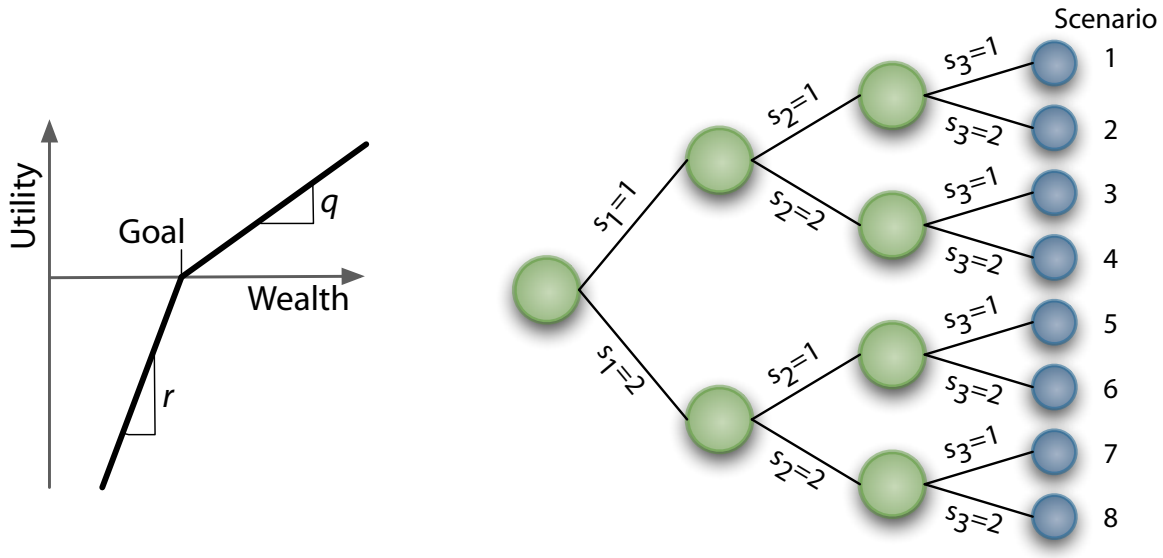
and $\boldsymbol{\xi}' \triangleq (\mathbf{q}', \mathbf{h}', \mathrm{vec}(\mathbf{T}))$ and the matrix $\mathbf{W}$ is assumed fixed. This problem is interpreted as follows:

- The play unfolds in two acts, separated by the disclosure of a random variable $\boldsymbol{\xi}$.†
- In the first stage (**decision-making**), the decision-maker must choose variables $\mathbf{x}$ to minimize a cost function made up of two parts: an immediate linear cost $\mathbf{c}'\mathbf{x}$ and an expected future cost $Q(\mathbf{x}, \boldsymbol{\xi})$ (that is only known during the second stage).
- In the second stage (**recourse**), the decision-maker has been revealed the random variable $\boldsymbol{\xi}$ and must act to minimize the consequences (second-stage cost function Q) of this state of affairs.

Hence, in the first stage, the decision-maker acts ahead of time *knowing that he will act optimally in the second stage* to make do as well as possible given the scenario that just occurred. If the space of the random variable $\boldsymbol{\xi}$ is discrete (finite number of scenarios), then the stochastic program (2.59) can be converted to a classical (albeit large) deterministic linear program.

*Not to be confused with dynamic programming or stochastic dynamic programming.

†The realization of this random variable is traditionally called a *scenario* in this context.



▲ **Figure 2.6.** ***Left:*** *Terminal utility function for the stochastic programming asset allocation example; at the end of the investment horizon, if the wealth goal is reached, subsequent investment can be made for a return of q% (e.g. in the risk-free rate), otherwise money must be borrowed at a rate of r%.* ***Right:*** *Scenario tree driving the model; at each period t, the joint stock–bond return is given by the scenario s_t. There is one decision node per period (large green circles), but decisions may depend on the entire realized history so far (i.e. the tree does not recombine). The smaller right-hand terminal nodes represent the final scenario outcomes.*

Extensions of the problem (2.59) to multiple decision stages are of course possible. To illustrate the application of the approach to asset allocation, we introduce a simple example inspired by Birge and Louveaux (1997). We assume an allocation tasks between N assets, $i = 1, \ldots, N$ over $t = 1, \ldots, T$ discrete periods. During each period t, a *scenario* s_t may occur, which is defined by the realization of random returns for all assets. More specifically, let $\xi(i, t, s_1, \ldots, s_t)$ be the random return for asset i during period t for scenario s_t, which may also depend on all previous realizations $s_1, \ldots, s_{t-1}$. This yields a (non-recombining) scenario tree illustrated in Fig. 2.6 (right). Furthermore, complete scenarios are given a probability $p(s_1, \ldots, p_T)$, which is used in the objective function (see below).

We assume that the investor is governed by the piecewise linear concave utility function shown in Fig. 2.6 (left). This function can be interpreted as follows: at horizon T the investor seeks to meet a financial goal G (for example, paying for Junior's college tuition). If this goal is met, the excess money can be invested at a yield of q%, but if not, the missing money must be borrowed at a rate of r%. Initially, the investor is endowed with W_0 dollars.

At the start of period t, the investor must make an allocation decision for each asset i, which is denoted $x(i,t,s_1,\ldots,s_{t-1})$, and represents the dollar amount invested in asset i for the duration of the period. The "identity" of the decision variables depend on the scenario history until that point, and corresponds to the larger nodes in the scenario tree of Fig. 2.6.

The objective function is the value of the utility function realized for each complete scenario, weighted by the probability of that scenario; since the utility is piecewise-linear, it is split out into two terms by means of "surplus variables" w and y corresponding, respectively, to borrowing at a rate of $r\%$ and investing at a yield of $q\%$ (the surplus variables are defined as function of terminal wealth through constraints, below),

$$\max \sum_{s_T} \cdots \sum_{s_1} p(s_1,\ldots,s_T)(-rw(s_1,\ldots,s_T) + qy(s_1,\ldots,s_T)).$$

The scenario probabilities $p(s_1,\ldots,s_T)$ are specified by the modeler. The constraint for the first period is to invest the totality of initial wealth,

$$\sum_i x(i,1) = W_0.$$

The middle-period constraints, for $t = 2,\ldots,T-1$, are budget balance constraints: the wealth invested during period t must be that resulting from the investment during period $t-1$ (accrued by the yields earned during that period), with no possible intermediate reinvestment,

$$\sum_i -\xi(i,t-1,s_1,\ldots,s_{t-1})\, x(i,t-1,s_1,\ldots,s_{t-2}) + \sum_i x(i,t,s_1,\ldots,s_{t-1}) \;=\; 0, \qquad \forall s_1,\ldots,s_{t-1}. \quad (2.60)$$

The last-period constraints take the terminal wealth generated in each scenario and split it among the surplus variables y and w for each scenario, depending on whether the accumulated wealth in that scenario is above or below the goal

$$\sum_i -\xi(i,T,s_1,\ldots,s_T)\, x(i,T,s_1,\ldots,s_{T-1}) - \; y(s_1,\ldots,s_T) + w(s_1,\ldots,s_T) \;=\; G, \qquad \forall s_1,\ldots,s_T. \quad (2.61)$$

We also force wealth to be positive in each period, along with the two surplus variables y and w,

$$\begin{aligned} x(i,t,s_1,\ldots,s_{t-1}) &\geq 0 && \forall i,t,s_1,\ldots,s_{t-1}, \\ y(s_1,\ldots,s_T) &\geq 0 && \forall s_1,\ldots,s_{t-1}, \\ w(s_1,\ldots,s_T) &\geq 0 && \forall s_1,\ldots,s_{t-1}. \end{aligned}$$

This completes the formulation of the multistage stochastic program for this (admittedly simplified) asset allocation example. As presented, the optimization problem can easily be transformed into a (deterministic) linear program, yet the method also handles path-dependent events such as transaction costs and taxes (since the scenario tree of Fig. 2.6 does not recombine).

Dantzig and Infanger (1993) discusses the solution of multiperiod portfolio problems in the stochastic programming framework, and present algorithms based on a Benders decomposition of the linear program and Monte Carlo importance sampling. A survey of stochastic programming approaches in finance is presented by Yu, Ji, and Wang (2003). The book edited by Zenios (1993) provides additional useful references.

Due to its ability to model the complex real-world dependencies, stochastic programming has been widely applied to the problem of *asset–liability management* where a portfolio does not only consist of investments (future incoming cash flows) but also liabilities (future outgoing cash flows; for instance faced by an insurer whose written policies represent liabilities to be paid in the future, and who has reserves to invest optimally). Dempster, Germano, Medova, and Villaverde (2003) provide an in-depth presentation of the theory of stochastic programming to this problem, and followed up with an application to the management of minimum guaranteed return funds (Dempster, Germano, Medova, Rietbergen, Sandrini, and Scrowston 2007). The books edited by Zenios and Ziemba (2006) and Dempster, Pflug, and Mitra (2008) contain recent material on this topic.

It is perhaps unfortunate that stochastic programming approaches to portfolio optimization have mostly been studied by the operations research community, and are relatively unknown to financial economists. One can argue that this may be attributable to two factors: first, it was only recently that solution algorithms and computational power have become sufficient to enable solution to large-scale problems; still, the technological hurdle to get even simple stochastic programming models working may remain prohibitive to some. Second, a traditional emphasis of the solution methods studied by financial economists has been on the characterization of compact optimal policies, with significant concern paid to analytical tractability. Stochastic programming solutions, on the other hand, are mostly numerical and do not necessarily convey as much insight into optimal behavior. Yet, there appears to be much opportunity to combine the potential of multiple approaches, for instance integrating the methodological maturity of single-period modeling (e.g. the expected-return and risk models of §2.1.7/p. 24) with the ability of stochastic programming to cleanly handle a large number of real-world investment constraints over multiple periods.

3 Machine Learning and Approximate Dynamic Programming

Everybody who is incapable of learning has taken to teaching.

— *Oscar Wilde*

THIS CHAPTER REVIEWS the most important concepts from statistical learning algorithms for understanding this thesis. We start with a review of useful definitions in machine learning (§3.1/p. 69), including the setting of supervised learning which is most employed here, useful function classes, and a discussion of the classical bias–variance trade-off. We then cover methods from the field of approximate dynamic programming and reinforcement learning (§3.2/p. 77), including a review of value and policy iteration and the method of temporal differences for approximate policy evaluation.

3.1 Useful Definitions in Machine Learning

3.1.1 Supervised Learning

We limit ourselves, in the present section, to the setting of supervised learning, more specifically that of regression (Ripley 1996; Hastie, Tibshirani, and Friedman 2001; Bishop 2006). Let $X \in \mathbb{R}^p$ and $Y \in \mathbb{R}$ be two random variables whose joint probability distribution $P(X, Y)$ is fixed but unknown. However, we have a set of N elements $D = \{(\mathbf{x}_i, y_i)_{i=1}^N\}$, called *training examples*, that are drawn IID from $P(X, Y)$.

The problem of supervised learning for regression can be stated as follows: we are seeking a function $f(X)$ to predict a value for Y given an X. More formally, let $L(y, \hat{y})$ a *loss function* that penalizes errors, where y is the observed "correct answer" and $\hat{y}$ is the predicted value (which we hope is close to y). We are seeking a function leading to the smallest *generalization error*,

$$C(f) = \mathbb{E}_{X,Y}[L(Y, f(X))], \tag{3.1}$$

where the expectation is taken over the joint distribution $P(X, Y)$. In theory, the function that best solves the problem of supervised learning is the one

that minimizes this generalization error,

$$f^* = \arg\min_{f \in \mathcal{F}} C(f), \tag{3.2}$$

where $\mathcal{F}$ represents the *function class* within which we restrict the search. We review below (§3.1.2/p. 71) a few important classes, including *linear models*, *artificial neural networks* and *kernel machines*.

It is of the utmost importance to emphasize that eq. (3.1) and (3.2) only provide theoretical definitions, and that *it is impossible to exactly compute generalization error* within the framework introduced above: the expectation in eq. (3.1) is taken with respect to the joint distribution $P(X, Y)$ that remains unknown, even though assumed to be fixed.

Loss Functions for Regression

The loss function $L(y, \hat{y})$ that is usually chosen for *regression* problems (in which case Y can assume continuous values within a subset of $\mathbb{R}$ rather than a finite number of discrete values as would be the case for a *classification* problem) is the *quadratic loss*,

$$L(y, \hat{y}) = (y - \hat{y})^2. \tag{3.3}$$

*See, for instance, Hastie, Tibshirani, and Friedman (2001).

It is easy to show* that the function minimizing quadratic loss is the same that computes the *conditional expectation* of Y given X, which is the "correct choice" of function for the vast majority of regression cases,†

$$\begin{aligned} C(f) &= \mathbb{E}_{X,Y}[(Y - f(X))^2] \\ &= \mathbb{E}_X \left[\mathbb{E}_{Y|X}[(Y - f(X))^2 \,|\, X] \right], \end{aligned}$$

and it suffices to take the value $f(x)$ which minimizes the inner expectation at each point x,

$$\begin{aligned} f(x) &= \arg\min_c \mathbb{E}_{Y|X}[(Y - c)^2 \,|\, X = x] \\ &= \arg\min_c \mathbb{E}_{Y|X}[Y^2 - 2cY + c^2 \,|\, X = x] \\ &= \mathbb{E}[Y \,|\, X = x]. \end{aligned}$$

Training and Test Sets

As mentioned above, it is impossible—except when making strong distributional assumptions—to find an exact solution to problem (3.2). We can

†There exists cases where a conditional median, or another quantile, is more appropriate; obviously, the circumstances of the application ultimately decide what target function should be learned.

only hope to find the best function $\hat{f}$ with respect to training set D, thereby minimizing the training error or *empirical loss*,

$$C_E(f) = \frac{1}{N}\sum_{i=1}^{N} L(y_i, f(\mathbf{x}_i)), \tag{3.4}$$

yielding

$$\hat{f} = \arg\min_{f\in\mathcal{F}} C_E(f). \tag{3.5}$$

Unfortunately, and this is perhaps one of the most fundamental aspects in the theory of statistical learning, the empirical loss is not a good estimator of generalization error: it is **optimistically biased**, in the sense that it underestimates the level of generalization error (Vapnik 1998). Intuitively, this arises because f is obtained as the result of a minimization, and it is an elementary result in statistics that for any set of IID random variables $\{Z_1, Z_2, \ldots\}$,

$$\mathbb{E}[\min(Z_1, Z_2, \ldots)] \leq \mathbb{E}[Z_i],$$

hence the bias.

To obtain an unbiased estimator of $C(f)$, we can compute the loss on a *test set*, disjoint from D, which is also drawn IID from the underlying distribution $P(X, Y)$. The obtained test error $\hat{C}(f)$ is, in this case, an unbiased estimator of $C(f)$ (but still exhibits variance, which depends on the size of the test set).

3.1.2 Function Classes

An optimization problem such as that of eq. (3.5) must normally be carried out within a well-defined domain, that of a *function class*. We shall consider function classes parameterized by a real vector $\boldsymbol{\theta} \in \mathbb{R}^P$: given a class and a fixed-size vector, we can construct a specific function.

Linear Functions

The class of linear functions is among the simplest. Given a parameter vector $\boldsymbol{\theta} = (\beta_0, \beta_1, \ldots, \beta_p)'$, the resulting function is computed as

$$f(\mathbf{x}; \boldsymbol{\theta}) = \beta_0 + \sum_{i=1}^{p} \beta_i \, \mathbf{x}_i. \tag{3.6}$$

The dependence of f on $\boldsymbol{\theta}$ may be made explicit, as above.

Given a training set $D = (\mathbf{X}, \mathbf{y})$ (where $\mathbf{X} \in \mathbb{R}^{N\times(p+1)}$ is the matrix of input variable values and $\mathbf{y} \in \mathbb{R}^N$ is the vector of target outputs), we can estimate $\boldsymbol{\theta} \in \mathbb{R}^p$ using, for instance, the well-known OLS* estimator,

*Ordinary Least Squares.

$$\hat{\boldsymbol{\theta}}^{\text{OLS}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}. \tag{3.7}$$

Under the hypothesis that the underlying functional relationship is linear and that the data is generated according to

$$\mathbf{y} = \mathbf{X}\boldsymbol{\theta} + \boldsymbol{\varepsilon},$$

where $\boldsymbol{\varepsilon} \overset{\text{iid}}{\sim} N(0, \sigma^2)$, then $\hat{\boldsymbol{\theta}}^{\text{OLS}}$ is the minimum-variance unbiased estimator of $\boldsymbol{\theta}$.*

Kernel Machines

A generalization of the linear model consists in expressing the solution as a sum of non-linear transformations of the training examples. We now assume a length-$N+1$ parameter vector, $\boldsymbol{\theta} = (\beta_0, \beta_1, \ldots, \beta_N)'$, where N is the number of training examples. The learned function is expressed as

$$f(\mathbf{x}; \boldsymbol{\theta}) = \beta_0 + \sum_{i=1}^{N} \beta_i K(\mathbf{x}, \mathbf{x}_i),$$

with the $\mathbf{x}_i$ typically training-set examples, and where the function $K(\cdot, \cdot)$, called a *kernel function*, is a continuous symmetric positive-definite function such that

$$\int_{\mathcal{X}\times\mathcal{X}} K(\mathbf{x}, \mathbf{z}) f(\mathbf{x}) f(\mathbf{z})\, \mathrm{d}\mathbf{x}\, \mathrm{d}\mathbf{z} \;\geq\; 0$$

for all $f \in L_2(\mathcal{X})$, with $\mathcal{X}$ a compact subset of $\mathbb{R}^p$. If these conditions on $K(\cdot, \cdot)$ are satisfied, it can be shown (Schölkopf and Smola 2001) that it corresponds to the weighted inner product of some mapping $\boldsymbol{\phi}(\cdot)$ (typically mapping $\mathbb{R}^p$ to a much higher-dimensional *feature space*),

$$K(\mathbf{x}, \mathbf{z}) = \sum_{j=1}^{\infty} \lambda_j \phi_j(\mathbf{x}) \phi_j(\mathbf{z}),$$

where $\phi_j(\cdot)$ is the function that maps to the j-th element of $\boldsymbol{\phi}(\cdot)$.

This representation is at the core of a bewildering variety of algorithms collectively known as *kernel machines*, all differing in the details of the training objective and algorithms,† and of which SVM‡s (Boser, Guyon, and Vapnik 1992) and Gaussian processes (Williams and Rasmussen 1996) are examples. The latter are used extensively in Chapter 6, where a more detailed review of the relevant algorithms is provided.

‡*Support Vector Machine.*

*This result is covered by introductory statistics and econometrics textbooks; e.g. Greene (2007).

†And, obviously, the resulting parameters β_i. A significant concern in this context is often to achieve *sparsity* where most β_i are zero; the training examples associated with the remaining non-zero parameters are termed *support vectors*.

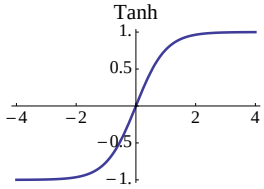
Artificial Neural Networks

Another generalization of the linear model is obtained by introducing one (or more) non-linear intermediate steps between input and output variables. Let the parameter vector be given by

$$\boldsymbol{\theta} = (\alpha_{0,0}, \ldots, \alpha_{H,p}, \beta_0, \beta_1, \ldots, \beta_H)'.$$

The function computed by a *feed-forward artificial neural network* with a single hidden layer of H units is

$$f(\mathbf{x}; \boldsymbol{\theta}) = \beta_0 + \sum_{i=1}^{H} \beta_i \tanh\left(\alpha_{i,0} + \sum_{j=1}^{p} \alpha_{i,j}\mathbf{x}_i\right). \tag{3.8}$$



*Multi-Layer Perceptron.

A general representation of such a network is given in Fig. 3.1. The intermediate function $\tanh(\cdot)$ introduces a nonlinearity in the model and constitutes the fundamental element enabling an MLP* to operate as a *universal approximator*; it can be shown that with enough hidden units H, the MLP can represent any continuous function defined on a compact support with an arbitrary precision (Hornik, Stinchcombe, and White 1989).

The parameter vector $\boldsymbol{\theta}$ must be estimated by numerical optimization by minimizing an empirical loss criterion, such as that of eq. (3.4). The most commonly used algorithm to compute the gradient of this loss function with respect to each parameter of the network is known as *error backpropagation* (Rumelhart, Hinton, and Williams 1986; Bishop 1995) and is an efficient application of the elementary chain rule of differential calculus. It is used jointly with classical unconstrained non-linear optimization algorithms, such as stochastic gradient descent (Benveniste, Metivier, and Priouret 1990) or conjugate gradients (Bertsekas 2000).

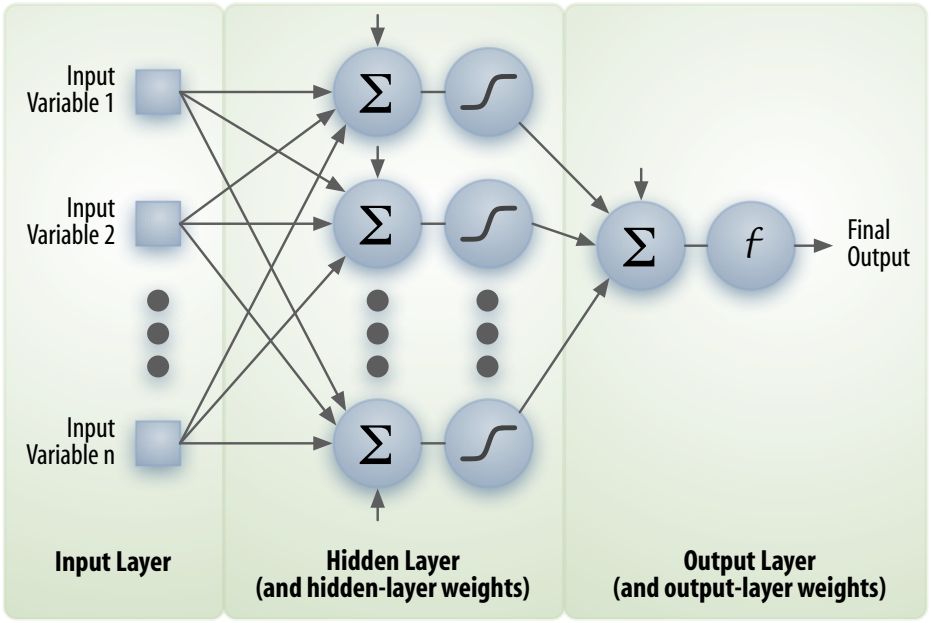
3.1.3 The Bias–Variance Trade-Off

For a quadratic loss criterion, it is possible to decompose the generalization error into explicit terms of *bias* and *variance*, which provide considerable intuition on the causes contributing to the overall error. Decompositions of this kind have been long known for regression in statistics, and were introduced in the context of neural networks by Geman, Bienenstock, and Doursat (1992).

In this section, we consider that the target y is a deterministic function $f(\cdot)$ of input variables $\mathbf{x}$, contaminated by an IID additive noise ϵ,

$$y = f(x) + \epsilon,$$

such that $\mathbb{E}[\epsilon] = 0$ and $\mathbb{E}[\epsilon^2] = \sigma^2$. The (random) function found by the learning process, $\hat{f}(\cdot)$, is that which minimizes the empirical mean-squared



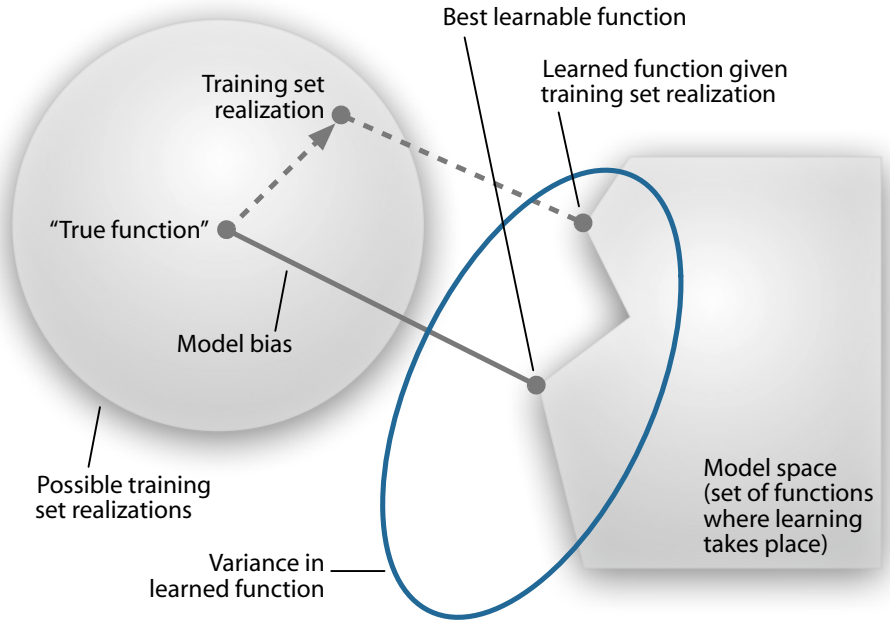
▶ **Figure 3.1.** *Illustration of a feed-forward neural network. The hidden-layer nonlinearities compute the* $\tanh(\cdot)$ *function. Each connection in the network has an associated multiplicative weight, part of the parameter vector* $\boldsymbol{\theta}$*. The output-layer function* f *depends on the application; it can be the identity for regression or the logistic for two-class classification.*

error on a random training set. The generalization mean-squared error for $\hat{f}(\cdot)$, for a given $\mathbf{x}$, is expressed as

$$\begin{aligned}
\mathbb{E}[(Y-\hat{f}(X))^2 \,|\, X=x] &= \mathbb{E}[(f(x)+\epsilon+\mathbb{E}\hat{f}(x)-\mathbb{E}\hat{f}(x)-\hat{f}(x))^2] \\
&= \mathbb{E}[(\epsilon+(f(x)-\mathbb{E}\hat{f}(x))+(\hat{f}(x)-\mathbb{E}\hat{f}(x)))^2] \\
&= \mathbb{E}[\epsilon^2]+\mathbb{E}[(f(x)-\mathbb{E}\hat{f}(x))^2] \\
&\quad +\mathbb{E}[(\hat{f}(x)-\mathbb{E}\hat{f}(x))^2] \\
&\quad \underbrace{-2\mathbb{E}[(f(x)-\mathbb{E}\hat{f}(x))(\hat{f}(x)-\mathbb{E}\hat{f}(x))]}_{\text{zero}} \\
&= \sigma^2+\underbrace{(f(x)-\mathbb{E}\hat{f}(x))^2}_{\text{bias}^2}+\underbrace{\mathbb{E}[(\hat{f}(x)-\mathbb{E}\hat{f}(x))^2]}_{\text{variance}}.
\end{aligned}$$

At a given point $\mathbf{x}$, the error is therefore a function of three components:

- The noise σ^2 at that point, which is irreducible;
- The (squared) bias of the function class, which is the distance between the "true" function $f(\cdot)$ and the best admissible function (in expectation) within the chosen function class $\mathbb{E}\hat{f}(\cdot)$.
- The variance in the actual function after training, representing the "average distance" between $\hat{f}(\cdot)$ (which is a function of the sampling noise affecting the particular training set used to construct $\hat{f}(\cdot)$) and the "average across all training sets" $\mathbb{E}\hat{f}(\cdot)$.



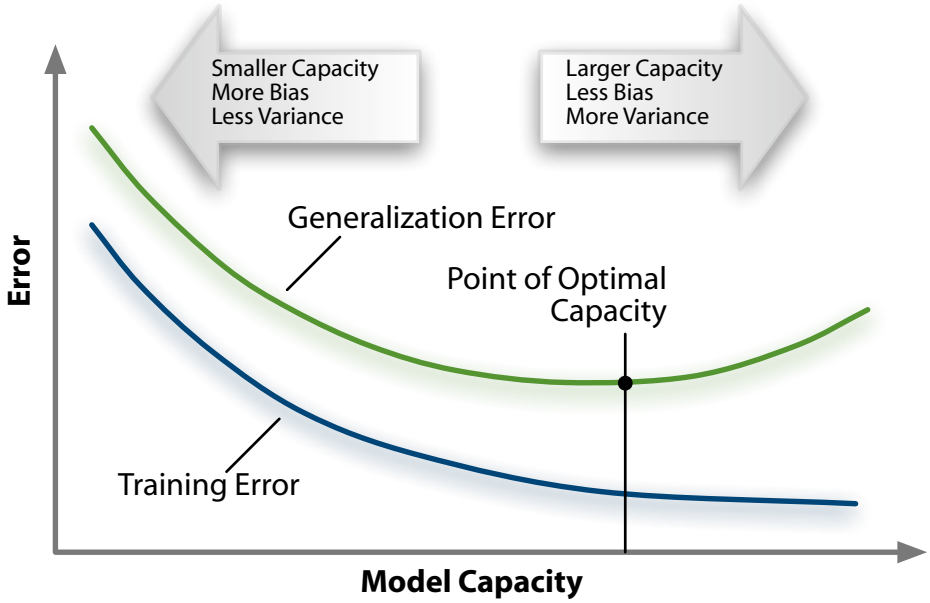
◀ **Figure 3.2.** *Decomposition of the generalization error (for a quadratic loss function) into* ***bias*** *and* ***variance****. (Illustration inspired by Hastie, Tibshirani, and Friedman (2001).)*

This decomposition of the generalization error is schematically illustrated in Fig. 3.2. The bias depends strictly on the chosen function class: on the one hand, a richer class (which contains more functions, and hence likely a function closer to the "true function") will have a lower bias; on the other hand, depending on the variations induced by the random sampling of the training set (which is of finite size and polluted by the noise ϵ), the learned function will try to match as closely as possible the *particular training set* used, and could be quite far away from the best function within the class, namely the one corresponding to the projection of the true underlying function $f(x)$ on the set of functions within which we restrict the search. This results in *variance*, which becomes worse as the function class gets richer.*

This fundamental compromise between the richness of a function class and estimation variance is called the *bias–variance trade-off*, schematized in Fig. 3.3. This illustration shows that as the function class becomes richer†, the training error decreases monotonically (assuming that parameter optimization is feasible for the function class). At the same time, the *generalization error* initially decreases, as a result of a bias decrease (since we are getting closer to the underlying "true function"), reaches a minimum point, and *starts to increase* again due to an increase in variance (we are fitting the noise in the training set). For each problem, there exists a point of optimal capacity giving the smallest generalization error, and representing the best trade-off between bias and variance. Unfortunately, although this point usually depends monotonically on the number of training examples N,

*Since a larger function class will more easily fit the noise realizations in a given training set.

†Which can be formalized using the notion of ***Vapnik–Chervonenkis dimension***; e.g. Vapnik (1998).



▶ **Figure 3.3.** *An increase in model capacity can always bring down the training error, whereas the generalization error starts to increase beyond a point of* ***optimal capacity****, which is specific to each problem.*

the training error is often of little help to establish its location. Two points become apparent:

▶ **1. Capacity Control :** It is imperative to control model capacity in order to get good generalization performance. This point shall be made repeatedly throughout Chapters 5 and 6, since the key to obtaining good out-of-sample performance in financial problems is frequently akin to a large-scale exercise in capacity control.

▶ **2. Optimal Capacity and Cross-Validation :** Several tools in machine learning (such as regularization) offer means to control model capacity, but are of no help for determining the point of optimal capacity. To this end, one may train a set of models and select the one giving the best performance on a separate *validation dataset.*[*] In a Bayesian setting, one may perform model selection by evaluating the *marginal likelihood* of the data for several distinct models and picking the one that best explains the data (i.e. exhibits the highest marginal likelihood; see §6.2.3/p. 251).[†]

A variant on the validation-set approach is called *cross-validation* (Stone 1974), which splits the training set D in K disjoint subsets $\{D_i\}$,

$$D = \bigcup_{i=1}^{K} D_i, \quad D_i \cap D_j = \emptyset, i \neq j.$$

[*] *Which is a disjoint set from the training and test sets, also assumed to be drawn i.i.d. from* $P(X, Y)$.

[†] Although it must be added that a "true Bayesian" would not be content with selecting a single model, but would instead integrate over possible models when making a decision.

The procedure repeats the following for each $i = 1, \ldots, K$,

1. Set aside the subset D_i for validation purposes.
2. Train a set of models (each with a different capacity) on the training set $D - D_i$.
3. Test each of the models trained in Step 2 on the validation set D_i, and save the result.

The selected model is that with the smallest average validation error, computed from the sample mean of the errors saved in Step 3. We note that the models are always trained on data that is different from what is used to compute the test performance, and assuming that the original data is indeed drawn IID from the underlying distribution, the computed test error is therefore an unbiased estimator of the generalization error.*

If the data is not IID but related by sequential dependencies, a variant of cross-validation called *sequential validation* can be used. This is described at length in Chapter 4.

3.2 Approximate Dynamic Programming

3.2.1 Classical Approximation Methods

Approximation methods for dynamic programming that make use of learning algorithms are also referred to as *reinforcement learning* or *neurodynamic programming* (Bertsekas and Tsitsiklis 1996; Sutton and Barto 1998; Si, Barto, Powell, and Wunsch 2004; Powell 2007). We give a very brief overview of some of these approaches in this section.

In order to simplify the presentation, we shall consider an infinite-horizon stochastic shortest-path problem without discounting† (Bertsekas 2007), on a finite set of states denoted $\mathcal{X} = \{1, 2, \ldots, n\}$, in addition to an absorbing terminal state, described below, denoted 0. In each state i, a control ("action") must be chosen from a finite set $U(i)$; we shall denote the union of all such sets by $\mathcal{U}$. In state i, choosing the control u governs the probability of making a transition to the next state j, $p_{ij}(u)$. Upon reaching the terminal state, we cannot leave it since it is absorbing; hence we have $p_{0j}(u) = 0, \forall u, \forall j \geq 1$. We assume that when a transition is taken, a reward $g(i, u, j)$ is received (which is stochastic, since it is function of the state j to which we are moving); moreover, all rewards from the terminal state are

*Nevertheless, this says nothing of the variance of this estimator, which can be high.

†*Discounting* refers to the notion that the present value of a future reward is less than the same nominal reward received today. This is generally achieved by introducing a *discount factor* γ, $0 < \gamma < 1$. The value at time t of a reward R_{t+k} to be received k time-steps later is equal to $\gamma^k R_{t+k}$.

zero. We shall also assume that the terminal state is reachable from all starting states within a finite number of transitions with probability one.

*We shall sidestep the technicalities requiring so-called "proper policies" that induce bounded value functions for all states.

We consider stationary deterministic policies $\mu(\cdot)$,* that are functions giving the control to pick in state i ($\in U(i)$). Once the policy set, the traversed state sequence $i_0, i_1, \ldots$ becomes a Markov chain with transition probabilities

$$P(i_{k+1} = j | i_k = i) = p_{ij}(\mu(i)). \tag{3.9}$$

Let N be the (stochastic) number of states in this chain before reaching the terminal state 0. The *value* $J^\mu(i)$ of a state i under a stationary policy μ is the expected reward incurred from this state until the terminal state is reached,

$$J^\mu(i) = \mathbb{E}\left[\sum_{k=0}^{N-1} g(i_k, \mu(i_k), i_{k+1}) \middle| i_0 = i \right]. \tag{3.10}$$

The function $J : \mathcal{X} \mapsto \mathbb{R}$ is called the *value function*.

We also define the **Q-value function** $Q^\mu(i, u)$ as the expected reward under the stationary policy μ, knowing that we initially start in state i and take action u, and thereafter follow policy μ,

$$Q^\mu(i, u) = \mathbb{E}\left[g(i_0, u_0, i_1) + \sum_{k=1}^{N-1} g(i_k, \mu(i_k), i_{k+1}) \middle| i_0 = i, u_0 = u \right]. \tag{3.11}$$

The function $Q : \mathcal{X} \times \mathcal{U} \mapsto \mathbb{R}$ is called the *Q-function*.

We shall denote by J^* and Q^* the value function and Q-value function under the *optimal policy* (assuming it is unique).

Bellman Equations

The value of state i under a given policy μ is subject to the *Bellman recurrence equation*,

$$\begin{aligned} J(i) &= \sum_j p_{ij}(\mu(i))(g(i, \mu(i), j) + J(j)) && (3.12) \\ &= (T_\mu J)(i), && (3.13) \end{aligned}$$

where the notation $T_\mu J$, commonly used in dynamic programming, is defined by eq. (3.12). This operator transforms a value function J into the value function at the following time-step under policy μ. Similarly, the value of state i under the *optimal policy* μ^* is subject to the following recurrence

$$\begin{aligned} J^*(i) &= \max_{u \in U(i)} \sum_j p_{ij}(u)(g(i, u, j) + J^*(j)) && (3.14) \\ &= (TJ^*)(i), && (3.15) \end{aligned}$$

where, as above, the operator T represents the value function at the following time-step under the *greedy policy* arising from J. Note that T_μ and T are

different operators: the former transforms a value function under policy μ, whereas the latter operates under the greedy policy arising from the value function. We shall use the notation T^k (resp. T_μ^k) to denote k repeated applications of operator T (resp. T_μ).

Value Iteration

Value iteration is the simplest, and oldest, algorithm to solve recurrence (3.14). It updates vector J as (Bertsekas 2007)

$$J(i) \leftarrow \max_{u \in U(i)} \sum_j p_{ij}(u)(g(i,u,j) + J(j)). \tag{3.16}$$

It can be shown that this iteration converges asymptotically towards J^* from any initial function J_0.

Policy Iteration

This is the second classical algorithm to solve the recurrence (3.14). Starting from an initial policy μ_0, it executes the following two steps at each iteration $k = 0, 1, \ldots$ of the algorithm,

1. Policy Evaluation We want to find the value function J^{μ_k} corresponding to the current policy μ_k, such that

$$T_{\mu_k} J^{\mu_k} = J^{\mu_k}. \tag{3.17}$$

To this end, we can iterate the operator T_{μ_k} in a way similar to value iteration, since for all J_0*

$$\lim_{i \to \infty} T_{\mu_k}^i J_0 = J^{\mu_k}.$$

Eq. (3.17) can also be solved directly for J^{μ_k} as a linear system, by substituting the definition of T_{μ_k} given by eq. (3.12). This can be achieved in time $O(n^3)$, where n is the number of states.

*And assuming that μ_k is a proper policy.

2. Policy Improvement From J^{μ_k}, we compute a new policy μ_{k+1} as

$$\mu_{k+1}(i) = \arg\max_{u \in U(i)} \sum_{j=0}^{n} p_{ij}(u)(g(i,u,j) + J^{\mu_k}(j)). \tag{3.18}$$

The above two steps are iterated until the new policy μ_{k+1} remains identical to the previous one μ_k. It can be shown that this algorithm converges in a finite number of iterations, which is usually relatively small (Bertsekas 2007).

3.2.2 Types of Approximation

Depending on our knowledge of the underlying *model* of the dynamical system, there exists three fundamentally different methods of making a decision (i.e. choosing a control) from a state i. Approximation methods can be used in all three cases.

Value Function Approximation

Assuming that we know the optimal value function $J^*(i)$ and we have a precise dynamical model, that is, we know the precise distribution of the rewards $g(i, u, j)$ of an action u in state i, as well as the transition probabilities $p_{ij}(u)$, then the optimal action is obtained as

$$u^*(i) = \arg\max_{u \in U(i)} \sum_j p_{ij}(u)(g(i, u, j) + J^*(j)). \tag{3.19}$$

In this case, we can choose a *function approximation of the value function* $\tilde{J}^*$ parameterized by a vector $\boldsymbol{\theta}$, which conceptually minimizes

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_i w_i \|J^*(i) - \tilde{J}^*(i, \boldsymbol{\theta})\|^2, \tag{3.20}$$

with $0 \leq w_i \leq 1, \sum_i w_i = 1$, where w_i is a weight assigned to state i and $\|\cdot\|$ is some norm (e.g. the Euclidean norm). This function can be used in the obvious manner to find the approximately optimal action,

$$\tilde{u}^*(i) = \arg\max_{u \in U(i)} \sum_j p_{ij}(u)(g(i, u, j) + \tilde{J}^*(j)). \tag{3.21}$$

Since in practice, we have no knowledge of $J^*(i)$, the result of eq. (3.20) cannot be implemented directly. We explain below (§3.2.3/p. 82 and following) how to directly find $\boldsymbol{\theta}$ without a prior knowledge of the optimal value function or explicit minimization of eq. (3.20), a costly proposition for large state spaces.

Q-Value Function Approximation

Absent detailed model knowledge, yet given knowledge of the Q-value function, the optimal action is given by

$$u^*(i) = \arg\max_{u \in U(i)} Q(i, u). \tag{3.22}$$

We can then approximate these Q-values using the **Q-Learning algorithm** (Watkins 1989; Watkins and Dayan 1992), whose details are omitted for brevity.

Policy Learning

Finally, we can attempt to directly learn the policy, namely to find a function $\mu(i;\boldsymbol{\theta})$ yielding the action to be taken in state i,

$$\tilde{u}^*(i) = \mu(i;\boldsymbol{\theta}). \tag{3.23}$$

Conceptually, assuming real-valued controls, the parameter vector $\boldsymbol{\theta}$ can be obtained by finding

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \sum_i w_i \|u^*(i) - \mu(i;\boldsymbol{\theta})\|^2, \tag{3.24}$$

where the optimal action $u^*(i)$ is given by eq. (3.19) and w_i is a weighting scheme satisfying the same conditions as for eq. (3.20). In practice, of course, the optimal action $u^*(i)$ is usually not readily available and one must resort to *policy gradient* approaches to optimize $\boldsymbol{\theta}$. We discuss these methods more deeply in §3.2.4/p. 86.

3.2.3 Linear Approximators

A first approximation architecture, whose roots can be traced back to Bellman and Dreyfus (1959), relies on a linear approximation of the value function. This is used jointly with eq. (3.19) to make decisions. It is generally used with features that are extracted from the current state.

Definition 6 (Feature) *A **feature** is a function $\phi : \mathcal{X} \mapsto \mathbb{R}$ that represents a "significant" aspect of a state $i \in \mathcal{X}$ as a real number.*

In general, we seek a feature representation of the value function that works well for a number of policies—not just a given policy—since the policy will constantly change as part of the learning process (e.g. within policy iteration).

Let $\{\phi_1, \ldots, \phi_M\}$ be a set of features. Given a state i, we define the *feature vector* corresponding to this state,

$$\phi(i) = (\phi_1(i), \ldots, \phi_M(i))'. \tag{3.25}$$

For a given problem, the features are usually chosen by hand according to the problem characteristics and engineering knowledge. Intuitively, we seek a feature vector that, for the majority of states, adequately summarizes the fundamental properties of the state towards approximating the value function under a given policy.

A linear approximator, given a feature vector $\phi(i)$ and parameters $\boldsymbol{\theta}$, computes an estimate of the value of state i as

$$\tilde{J}(i;\boldsymbol{\theta}) = \boldsymbol{\theta}'\phi(i). \tag{3.26}$$

The parameters $\boldsymbol{\theta}$ can be learned using a number of approaches; we briefly summarize the most common ones: approximate value iteration, Monte Carlo estimation, and temporal differences.

Approximate Value Iteration

In its original form, approximate value iteration was introduced by Bellman and Dreyfus (1959). Starting from an approximate value function $\tilde{J}(\cdot;\boldsymbol{\theta}_0)$ given by an initial parameter vector $\boldsymbol{\theta}_0$, the method selects, during iteration k, a representative subset of states $S_k \subseteq \mathcal{X}$ and computes one step of operator T for all states $i \in S_k$,

$$\hat{J}_{k+1}(i) = \max_{u \in U(i)} \sum_j p_{ij}(u)(g(i,u,j) + \tilde{J}(j;\boldsymbol{\theta}_k)). \tag{3.27}$$

It then updates the parameter vectors corresponding to $\hat{J}_{k+1}$ by minimizing (for instance) a quadratic cost criterion

$$\boldsymbol{\theta}_{k+1} = \arg\min_{\boldsymbol{\theta}} \sum_{i \in S_k} w_i \|\hat{J}_{k+1}(i) - \tilde{J}(i;\boldsymbol{\theta})\|^2, \tag{3.28}$$

where, as above, w_i is a weight assigned to state i. The set S_k can be chosen from *a priori* problem knowledge, or from the states visited under a given policy. For instance, one can use the greedy policy induced by $\tilde{J}(i;\boldsymbol{\theta}_k)$ or an ϵ-greedy policy (Sutton and Barto 1998) that chooses the greedy action with probability $1-\epsilon$ and a random action with probability ϵ.

The advantage of ϵ-greedy policies is to provide a way to balance **exploration** and **exploitation**, a classical problem in control and reinforcement learning. A purely greedy policy ($\epsilon = 0$) exploits the current knowledge about the value function to choose the best action under the current policy; however, in doing so, it assumes that the estimated value function is perfect and that an alternative action choice would never yield a better outcome (even after learning). If the current policy is significantly different from the optimal one, it can remain stuck in local optima since actions yielding greater rewards (possibly several steps ahead) are never attempted. In contrast, with $\epsilon > 0$, a random action is sometimes chosen instead of the greedy one, and this ensures that some exploration of "worse actions" remain. Obviously, too high a value of ϵ (given the quality of the current policy compared to the optimal one) causes learning to be very slow due to the noise introduced in the value function back ups.

▶ **Performance Bounds :** It is possible to establish performance guarantees for this algorithm (Bertsekas and Tsitsiklis 1996, p. 332). In particular, assume that the approximation architecture is flexible enough and that states can be sampled sufficiently densely so that

$$\|\tilde{J}_{k+1} - T\tilde{J}_k\|_\infty \leq \varepsilon \tag{3.29}$$

where $\tilde{J}_k \triangleq \tilde{J}(\cdot;\boldsymbol{\theta}_k)$ is the cost function after k iterations of the algorithm and $\|\cdot\|_\infty$ is the infinity-norm. We can establish that $\|T^k J_0 - \tilde{J}_k\|_\infty \leq k\varepsilon$ for all initial value functions J_0. By induction on k, we note that this is true

for $k = 0$; now assuming that this is true for some $k \geq 0$, at step $k+1$ we have

$$\begin{aligned}\|T^{k+1}J_0 - \tilde{J}_{k+1}\|_\infty &\leq \|T^{k+1}J_0 - T\tilde{J}_k\|_\infty + \|T\tilde{J}_k - \tilde{J}_{k+1}\|_\infty \\ &\leq \|T^k J_0 - \tilde{J}_k\|_\infty + \varepsilon \\ &\leq k\varepsilon + \varepsilon,\end{aligned}$$

which implies that the distance (in the infinity-norm sense) between the approximate value function $\tilde{J}_k$ after k steps and the "exact" k-step value function, $J_k \equiv T^k J_0$, does not grow worse than linearly in k. In other words, our error in approximating J_k by $\tilde{J}_k$ remains bounded. Let N be such that $\|T^N \tilde{J}_0 - J^*\|_\infty \leq \delta$ (this N exists due to the convergence of value iteration), the above inequality implies that the distance to the optimal value function, after N steps, can be bounded by

$$\|\tilde{J}_N - J^*\|_\infty \leq \|\tilde{J}_N - T^N J_0\|_\infty + \|T^N J_0 - J^*\|_\infty \leq N\,\varepsilon + \delta.$$

Better results can be obtained for problems with discounting. We introduce a discounting factor that reduces the perceived value of a future state. Let γ be the discounting factor. It can be shown that convergence to a function "quite close" to the optimal value function J^* is guaranteed, with

$$J^* - \frac{\varepsilon}{1-\gamma}\boldsymbol{\iota} \leq \lim\inf_{k\to\infty} J_k \leq \lim\sup_{k\to\infty} J_k \leq J^* + \frac{\varepsilon}{1-\gamma}\boldsymbol{\iota}, \tag{3.30}$$

where $\boldsymbol{\iota}$ is a vector of ones of the same size as the state space.

These bounds may appear slightly disappointing (linear in k for problems without discounting, and $O((1-\gamma)^{-1})$ for problems with discounting); however, their very existence indicates that this approximation algorithm is, in essence, well founded.

Monte Carlo Policy Evaluation for Tabular Representations

We now consider the first of two approaches to approximate the **policy evaluation** step of the policy iteration algorithm (p. 79). Rather than solving a system of linear equations (whose size is the cardinality of the state space), we approximate its solution by computing, from each state, the cumulative reward accrued under a large number of *simulated trajectories*. To this end, we assume that a model of the underlying dynamical system is available: that, from a given state i and control u, we can draw a next state j from the distribution $p_{ij}(u)$ as well as the resulting reward $g(i,u,j)$. The advantage of this approach is that it is often easier to draw samples from the distribution than to have an explicit representation of $p_{ij}(u)$.

From an initial state i_0^m and given a fixed policy μ, we draw a set of trajectories $(i_0^m, i_1^m, \ldots, i_N^m)$, where $i_N^m = 0$ (the absorbing terminal state), and $m = 1, \ldots, K$ is the trajectory index. For each trajectory, we compute the total reward to reach the terminal state,

$$c(i_0^m) = g(i_0^m, \mu(i_0^m), i_1^m) + \cdots + g(i_{N-1}^m, \mu(i_{N-1}^m), i_N^m). \tag{3.31}$$

The value function estimated for each state is simply the sample average, where we assume here that the value is explicitly stored for each state in the form of a table,

$$\hat{J}(i) = \frac{1}{K} \sum_{m=1}^{K} c(i^m). \tag{3.32}$$

We can also construct an on-line estimator of the value function J, which is updated after each simulated trajectory,

$$\hat{J}(i) \leftarrow \hat{J}(i) + \alpha_m (c(i^m) - \hat{J}(i)), \tag{3.33}$$

where α_m is a learning rate, usually a function of the iteration number.* The procedure may be initialized with $\hat{J}(i) = 0$.

Temporal Differences for Tabular Representations

The method of temporal differences is a generalization of Monte Carlo policy evaluation introduced by Richard Sutton (Sutton 1984; Sutton 1988). It is most easily understood by rewriting the state value update of eq. (3.33), for a given trajectory, as†

†*Assuming, as previously, a tabular representation of the value function $\hat{J}$.*

$$\begin{aligned}\hat{J}(i_k) \leftarrow \hat{J}(i_k) + \alpha \Big(& \Big(g(i_k, \mu(i_k), i_{k+1}) + \hat{J}(i_{k+1}) - \hat{J}(i_k) \Big) \\ & + \Big(g(i_{k+1}, \mu(i_{k+1}), i_{k+2}) + \hat{J}(i_{k+2}) - \hat{J}(i_{k+1}) \Big) \\ & + \quad \cdots \\ & + \Big(g(i_{N-1}, \mu(i_{N-1}), i_N) + \hat{J}(i_N) - \hat{J}(i_{N-1}) \Big) \Big).\end{aligned}$$

This update can be simplified as

$$\hat{J}(i_k) \leftarrow \hat{J}(i_k) + \alpha \left(d_k + d_{k+1} + \cdots + d_{N-1} \right), \tag{3.34}$$

where the *temporal difference* (TD) d_k is given by

$$d_k \triangleq g(i_k, \mu(i_k), i_{k+1}) + \hat{J}(i_{k+1}) - \hat{J}(i_k). \tag{3.35}$$

Intuitively, this TD represents the difference between the future reward estimator for a given state, $\hat{J}(i_k)$ and the future reward based on a one-step simulated trajectory, $g(i_k, \mu(i_k), i_{k+1}) + \hat{J}(i_{k+1})$. The temporal difference is a sample of the "Bellman error" in eq. (3.12).

*For instance, $\alpha_m = \frac{1}{m}$; the usual conditions for stochastic approximation, namely $\sum_m \alpha_m = \infty$ and $\sum_m \alpha_m^2 < \infty$ must be satisfied to guarantee the convergence of $\hat{J}(i)$ to J^μ as $m \to \infty$. See, e.g., Benveniste, Metivier, and Priouret (1990).

▶ **The TD(λ) Algorithm :** This algorithm assigns exponentially-decreasing weights to temporal differences when updating the value of a state i_k. A possible state-value update rule takes the form

$$\hat{J}(i_k) \leftarrow \hat{J}(i_k) + \alpha \sum_{m=k}^{\infty} \lambda^{m-k} d_m, \tag{3.36}$$

where $d_m \equiv 0, m \geq N$, and α is a learning rate (which can be a function of k).

If $\lambda = 1$, the special case of the Monte Carlo estimator is recovered.

The case $\lambda = 0$ implements the TD(0) algorithm, one of the first-proposed methods of reinforcement learning (Sutton 1988). Its expectation is equal to the Bellman equation.

Literally dozens of variants of the update (3.36) have been proposed. Some operate in an on-line fashion and incorporate the impact of a new temporal difference on all past states previously encountered along the trajectory (an idea known as *eligibility traces*); other methods impose hard limits on the temporal span of a TD; etc. A comparison between some of those approaches is provided by Sutton and Barto (1998).

▶ **Convergence Guarantees :** It can be shown that for tabular representations of the value function $\hat{J}$, nearly all variants of policy evaluation by temporal differences converge to the true function J^μ, both for finite- and infinite-horizon problems, with or without discounting (Bertsekas and Tsitsiklis 1996).

▶ **Policy Improvement :** Given an estimator $\hat{J}$ of the value function J^μ under policy μ, computed from the algorithm TD(λ), one can produce a new policy by following the usual policy improvement step of the policy iteration algorithm (eq. (3.18)).

Temporal Differences for Non-Tabular Representations

The temporal difference algorithm is also well suited to a parametric approximation of the value function, for instance using the linear architecture introduced previously or a feed-forward neural network (§3.1.2/p. 73). The only condition to use such architectures is to be able to compute the *gradient* of the value function at a given state with respect to model parameters $\boldsymbol{\theta}$. For linear approximators, this gradient is trivial to compute, and can be efficiently computed by backpropagation in the case of neural networks.

In an "off-line" version of the TD(λ) algorithm, we start by sampling a trajectory $i_0, \ldots, i_N$, and then update the parameters $\boldsymbol{\theta}$ via a gradient ascent step weighted by all temporal differences,

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha \sum_{m=0}^{N-1} \nabla \tilde{J}(i_m, \boldsymbol{\theta}) \sum_{k=m}^{N-1} d_k \, \lambda^{k-m}, \tag{3.37}$$

where, as above, α is a learning rate. On-line versions of this algorithm have also been introduced.

▶ **Convergence Guarantees :** For approximation architectures, the performance guarantees available for the TD(λ) architecture are much weaker than for tabular representations. Several examples of *divergences* have been shown for TD(λ) with non-linear architectures, as well as for TD(0) (including for linear architectures) if the states are not sampled under the policy μ (a property known as "off-policy learning").

In the general case, the best guarantees can be established for TD(1), even if this is the version exhibiting the slowest practical convergence due to large estimation variance. Moreover, substantial results can be established for linear architectures under some conditions (Tsitsiklis and Roy 1997; Tsitsiklis and Roy 2001). The convergence properties of approximation architectures for reinforcement learning remain an area of active study.

3.2.4 Direct Policy Learning and Actor-Critic Methods

The approximation methods covered so far focused on learning a representation of the value function (or the related Q-values), which represent the expected future reward* of following a given policy from a starting state. From an approximation of the value function, a *greedy policy* is obtained by choosing the action maximizing the next-state value in any given state.

However, these approaches suffer from several weaknesses. First, by employing a greedy policy (or an ϵ-greedy policy which would allow some exploration), they do not readily lend themselves to representing *stochastic policies* that probabilistically select among the possible actions in proportion to their desirability.† Second, they tend to yield *unstable policies*, since small changes in the estimated value function can produce rank changes among possible actions, leading to discontinuities in the resulting policy.‡ Finally, they constitute an indirect means of solving the problem that is usually of interest—that of finding an optimal policy—and one could hope that sidestepping this issue would lead to faster convergence.

Policy learning methods try to directly represent the policy by a parameterized function approximator, and estimate parameter values that exhibit good performance. Denote the stochastic policy followed in state i under parameters $\boldsymbol{\theta}$ by $\mu(i;\boldsymbol{\theta})$; we assume that this function yields a probability distribution over possible actions in $U(i)$. Writing $\mu(i,u;\boldsymbol{\theta})$ yields the probability of action u in state i.

*Or *cost-to-go*, if one minimizes costs instead of maximizing rewards.

†In some contexts, it is known that stochastic policies are superior to deterministic ones; see, e.g., Neyman and Sorin (2004) or Bagnell, Kakade, Ng, and Schneider (2004) for applications to POMDPs.

‡A number of divergence results arising from these instabilities have been documented in the literature; see Bertsekas and Tsitsiklis (1996) for examples.

In the *policy gradient* method of Sutton, McAllester, Singh, and Mansour (2000), which is defined for discounted infinite-horizon problems (and non-discounted average cost-per-stage problems), one defines an objective function $\rho(\boldsymbol{\theta})$, which can be taken in our context to be the value function from one designated start state i_0, $\rho(\boldsymbol{\theta}) \equiv J^{\mu(\boldsymbol{\theta})}(i_0)$. Updates to the parameters are made by simple gradient ascent,

$$\boldsymbol{\theta}^{(\tau)} \longleftarrow \boldsymbol{\theta}^{(\tau-1)} + \alpha \frac{\partial \rho(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}},$$

where α is a positive step size and $\boldsymbol{\theta}^{(\tau)}$ represents the parameter vector during iteration τ of the algorithm; $\boldsymbol{\theta}^{(0)}$ is given. Assuming discrete state and action spaces, Sutton, McAllester, Singh, and Mansour (2000) show that the policy gradient can be written as

$$\frac{\partial \rho(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \sum_i d^{\mu}(i) \sum_u \frac{\partial \mu(i,u;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} Q^{\mu}(i,u), \tag{3.38}$$

where d^{μ} represents the stationary distribution of states under $\mu(\boldsymbol{\theta})$ (which is assumed to exist and is independent of i_0 for all policies) and $Q^{\mu}(i,u)$ is the Q-value of action u in state i under policy $\mu(\boldsymbol{\theta})$. The essential attribute of eq. (3.38) is that the gradient contains no term of the form $\partial d^{\mu}(i)/\partial \boldsymbol{\theta}$; in other words, the gradient is not affected by the effects of the policy change (as we optimize) on the stationary state distribution. This implies that the inner term can be estimated by sampling. For instance, if one were to use the actual rewards to estimate the $Q^{\mu}(i,u)$ factors, one recovers the REINFORCE algorithm of Williams (1992).

Sutton *et al.*'s other contribution is to show that the gradient expression (3.38) remains valid if one uses a parametric approximation of the Q-values, denoted $\tilde{Q}(i,u;\boldsymbol{\phi})$, where $\boldsymbol{\phi}$ is the associated parameter vector, as long as a "compatibility" condition is satisfied between the two function approximators,

$$\frac{\partial \tilde{Q}(i,u;\boldsymbol{\phi})}{\partial \boldsymbol{\phi}} = \frac{\partial \mu(i,u;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{1}{\mu(i,u;\boldsymbol{\theta})}.$$

In this case, the $Q^{\mu}(i,u)$ factors in eq. (3.38) can be substituted by $\tilde{Q}(i,u;\boldsymbol{\phi})$ and the policy gradient computation remains valid. Similar ideas were considered, in various contexts including POMDP*s, by Glynn (1986), Jaakkola, Singh, and Jordan (1995), Marbach and Tsitsiklis (2001) and Marbach and Tsitsiklis (2003).

*Partially Observed Markov Decision Process.

Methods based on the separate approximation of the policy and the value function are called *actor-critic algorithms*; the "actor" is the parameterized policy, and the "critic" (the value function approximation) is used to update the actor's parameters in a direction of policy improvement. They are studied in depth, including a number of convergence results, by Konda and Tsitsiklis (2003).

A simulation-based method, the PEGASUS algorithm of Ng and Jordan (2000), is based on Monte Carlo realizations of trajectories to approximate a value function (whose evaluation only under the starting state distribution plays a part in the performance criterion), followed by a gradient-based algorithm to improve the policy. The authors obtain a number of strong convergence results, in particular that as long as the number of Monte Carlo trajectories is at most polynomial in significant problem quantities (Vapnik-Chervonenkis dimension of the policy function class, maximum reward, discount factor, approximation quality), then a uniform convergence result of the value function approximation can be obtained, and this can be used to give guarantees on the performance of approximate policies. The key to making this algorithm work is the use of the *common random numbers* variance reduction method frequently used in stochastic simulation (Law and Kelton 2000). Ng, Kim, Jordan, and Sastry (2004) demonstrate impressive performance results of the PEGASUS algorithm on a helicopter control task.

A different family of approaches, based on population and evolutionary computation techniques, are covered in Chang, Fu, and Marcus (2007). These can be seen to supplement gradient-based approaches when the latter are not feasible.

Finally, a comparison between policy gradient and value function approximation methods is provided by Beitelspacher, Fager, Henriques, and McGovern (2006). On a difficult game-playing navigation problem, the use of a value-based algorithm (SARSA(λ); see Sutton and Barto 1998) is found to ultimately lead to better policies than the online policy gradient algorithm of Weaver and Tao (2001), as the latter appears prone to local optima.

4 Training Graphs of Learning Modules for Sequential Data

If you torture the data long enough, it will confess.

— *Ronald Coase*

Statistical learning algorithms have long found application in sequential decision-making tasks, particularly in economic and financial problems (Abu-Mostafa, Atiya, Magdon-Ismail, and White 2001; Shadbolt and Taylor 2002). In practical systems, individual learning algorithms are almost never used in isolation; rather, *complex networks* of modules* are designed that together provide the requisite functionality. For instance, in the context of financial portfolio management, complex investment strategies can result from the combination of learning-driven decisions, fixed decision rules (often used as safeguards), and both fixed and adaptive data preprocessing.

It is a challenge to rigorously and efficiently evaluate the performance of such "learning networks" (to be precisely defined below), especially with respect to criteria that take into account the whole sequence of decisions, such as financial performance measures (accounting for trading costs) in a portfolio management task. Due to the danger of *overfitting*, and associated *data snooping biases*, the vast majority of the machine learning literature (and more recently, the empirical finance literature†), provides, as a matter of course, out-of-sample evaluation of proposed models. Most of these take the form of a simple "train–test" split (also called an "estimation–validation" split), where an initial training set is used to fit model parameters, and a held-out test portion used to simulate post-deployment performance.

Unfortunately, for *sequential* decision-making tasks, single train–test splits exhibit several troublesome issues. First is the traditional question of *where to split*, with the objective of giving enough training data to train

*Where each module can be a complete learning algorithm in the traditional sense, rather than, e.g. a single hidden unit in a neural network.

*A version of this chapter has been submitted for publication as (Chapados and Bengio 2009b).

†Overfitting and data snooping biases were brought to the attention of the financial econometrics community by Lo and MacKinlay (1990). White (2000) has proposed a "reality check" test to account for the effects of biases attributable to repeatedly using the same data to construct several models, although some studies have reported that the test lacks the power to distinguish between "good" and "bad" models in some important cases (Hansen and Lunde 2005).

accurate models, and enough test data to derive low-variance performance estimates.*

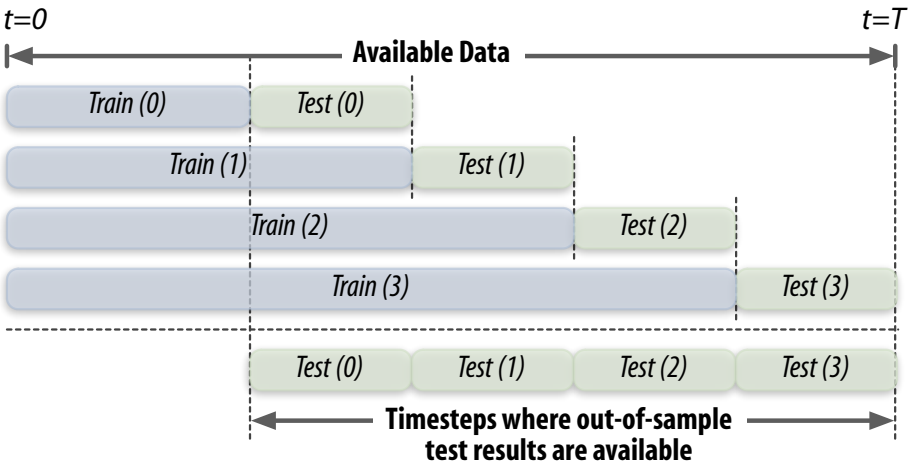
The second and deeper problem is related to *inherent non-stationarities* in the data distribution: many tasks exhibit variables whose distribution change significantly through time. In macroeconomic forecasting, this may take the form of *structural breaks*, caused by, as observed by Clements and Hendry (1999, p. xxiii), evolving economies that "are subject to sudden shifts, precipitated by changes in legislation, economic policy, major discoveries, and political turmoil". Stock and Watson (1996) document the evidence of structural instability in 76 representative post-War U.S. macroeconomic time series. In the finance literature, non-stationarities were reported for volatility in the form of (G)ARCH effects (Engle 1982; Bollerslev 1986; Campbell, Lo, and MacKinlay 1997); more recently, non-stationarities have also been recognized for expected returns and their effects on realized returns (Fama and French 2002). It is obvious that ultimately, the whole question of non-stationarity hinges on the choice of model: the prism through which one analyzes the world may admit sufficient inherent complexity to explain phenomena for which a simpler apparatus would be misspecified. Nevertheless, our present goal is not to carry out an exhaustive overview of evidence in favor of non-stationarity in any particular domain, but to defensively assume that some of those effects can occur, and examine how one should deal with them.

The third problem is that single train–test performance evaluation occults the reality faced by a flesh-and-blood decision maker: no rational person would train a computer model on a limited subset of data and then let this model run for an arbitrary period of time—without as much as an update to the model even as new data becomes available. Rather, one would rationally want to quickly make use of all available information, promptly updating the model to reflect new data.

Sequential validation (Bengio 1997; Gingras, Bengio, and Nadeau 2002; Chapados and Bengio 2001) is an empirical testing procedure that aims to emulate the behavior of said rational decision maker. Inspired by the well-known technique of cross-validation (Stone 1974; Hastie, Tibshirani, and Friedman 2001), its use is appropriate when the elements of a dataset cannot be permuted freely, such as is the case in sequential learning tasks. One can intuitively understand the procedure from the illustration in Fig. 4.1. One trains an initial model from a starting subset $\mathit{Train}(0)$ of the data,† which is tested out-of-sample on a data subset $\mathit{Test}(0)$ immediately following the end of the training set. This test set is then added to the training set for the

*This point is not so inconsequential as appears at first glance: anecdotically, we encountered more than once the scenario where an implementation of a model proposed in a paper worked quite well on the train–test split used in the paper, only to fail disgracefully when tested on any other period: this can be seen as an instance of the so-called *dataset selection* bias.

†"Starting" is meant in the temporal sense.



◀ **Figure 4.1.** *Illustration of* ***sequential validation****. A model is retrained at regular intervals, each time tested on a small subset of data that immediately follows (in temporal order) the end of the training set. At each iteration, the test data from the previous iteration is added to the training set.*

next iteration, a new model is trained and tested on a subsequent test set, and so forth. As the figure shows, at the end of the procedure, one obtains out-of-sample test results for a large fraction of the original data set, all the while always testing with a model trained on "relatively recent data".

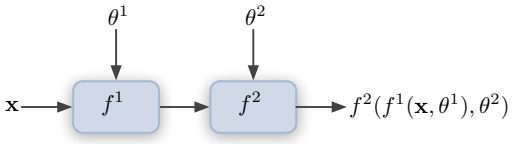
Stock and Watson (1999) use this technique in an economic forecasting context, calling it a "simulated out-of-sample forecasting methodology". They note that "this methodology provides a degree of protection against overfitting and detects model instability." In statistics, this methodology is known as *prequential analysis*, short for "predictive–sequential" (Dawid 1984; Dawid 1992).

One sees immediately that, as opposed to a single "train–test" split, sequential validation is not only a very effective way of making the most of limited (temporal) data; it also provides an approach to deal with what we can call—informally—"slow" non-stationarities, i.e. progressive changes in the underlying generating process distribution.* It also applies, contrarily to cross-validation, to contexts where a single unbroken *sequence of decisions* must be maintained.

4.1 Challenges with Sequential Validation

As shown in Fig. 4.1, sequential validation entails repeated interleaved steps of training and testing. When dealing with complex *networks of composed learning algorithms*, the training step itself involves the creation of a training set for each adaptive element in the network. To see how this may arise,

*If such non-stationarities are suspected, one should ensure that the training set remains of limited size—discarding old data—as the sequential validation proceeds, so as to at least ensure that distributionally different data eventually leaves the training set.



◀ **Figure 4.2.** *Elementary composition of function approximators $f^1(\cdot,\theta_1)$ and $f^2(\cdot,\theta_2)$. The input vector is $\mathbf{x}$; θ_1 and θ_2 represent learned parameter vectors.*

consider the very simple composition of two function approximators, $f^1(\cdot,\theta_1)$ and $f^2(\cdot,\theta_2)$, shown in Fig. 4.2. To unify notation, we shall denote by $\mathcal{T}_i^j$ and $\mathcal{U}_i^j$ respectively the *training* and *test* sets used for learner j at iteration i of sequential validation. Within a single iteration i of sequential validation, this network of two elements can be trained according to the following steps:

1. Construct the training set $\mathcal{T}_i^1$ for f^1
2. Train f^1 on $\mathcal{T}_i^1$ to obtain θ_i^1
3. Construct the training set $\mathcal{T}_i^2$ for f^2 by computing the outputs of $f^1(\cdot,\theta_i^1)$ on the examples within $\mathcal{T}_i^1$; this is the **key compositional step** for training*
4. Train f^2 on $\mathcal{T}_i^2$ to obtain θ_i^2

Across consecutive iterations of sequential validation, one can clearly anticipate some computational savings: for instance, the training set $\mathcal{T}_{i+1}^1$ has grown from $\mathcal{T}_i^1$ only by adjoining the elements in the corresponding iteration-i test $\mathcal{U}_i^1$; it does not need to be reconstructed from scratch. For $\mathcal{T}_{i+1}^2$, the situation is more unfortunate: in ordinary circumstances, no such shortcut exists for obtaining it from $\mathcal{T}_i^2$: since the behavior of $f^1(\cdot,\theta_i^1)$ must be assumed to differ arbitrarily from that of $f^1(\cdot,\theta_{i+1}^1)$ due to the difference in parameter vectors, the training set $\mathcal{T}_{i+1}^2$ would not generally be a strict expansion of $\mathcal{T}_i^2$—one needs to carry out step 3 in the above procedure in its entirety with the new θ_{i+1}^1 to construct $\mathcal{T}_{i+1}^2$. However, all is not lost: suppose that f^1 is non-adaptive and carries out fixed preprocessing; this implies that $\mathcal{T}_{i+1}^2$ becomes, in this case, a strict expansion of $\mathcal{T}_i^2$ and can be computed incrementally.

One might wonder why we appear to belabor this training-set incrementality issue. It turns out that for complex real-life networks constituted of a large number of arbitrarily-composed elements, the training-set construction steps may represent a significant fraction of the total running time of a simulation, particularly when frequent retrainings are requested in the se-

*Note that this step only provides the *input portion* for $\mathcal{T}_i^2$; the targets (desired outputs) need to be specified separately. In most applications of concern to us, this should not constitute a difficulty. For instance, if f^2 is to act as a predictor of asset returns, the target returns can be computed independently.

quential validation.* In addition, several learning algorithms can operate much more efficiently when given an "incremental" training set, such as the so-called *recursive estimators* for linear regression (Greene 2007).

Furthermore, in practical applications, one may wish to compose, not two or three elements, but several dozens—even hundreds—of them; where an illustration, depicted in Fig. 4.3, is further studied in §4.6/p. 114. In these cases, it becomes immensely time-consuming and error-prone for a human to identify by hand which elements can have their training set computed incrementally (from one iteration of sequential validation to the next), and which ones require full computation. To compound the difficulty, as we shall show below, it is sometimes not possible with economic time-series data to make this choice **statically** by just examining the graph of interconnections between elements; it may occur that the "status" of an element (namely, whether its training set admits incremental updates or must be reconstructed from scratch) changes along the simulation.

4.1.1 Goals and Organization of this Chapter

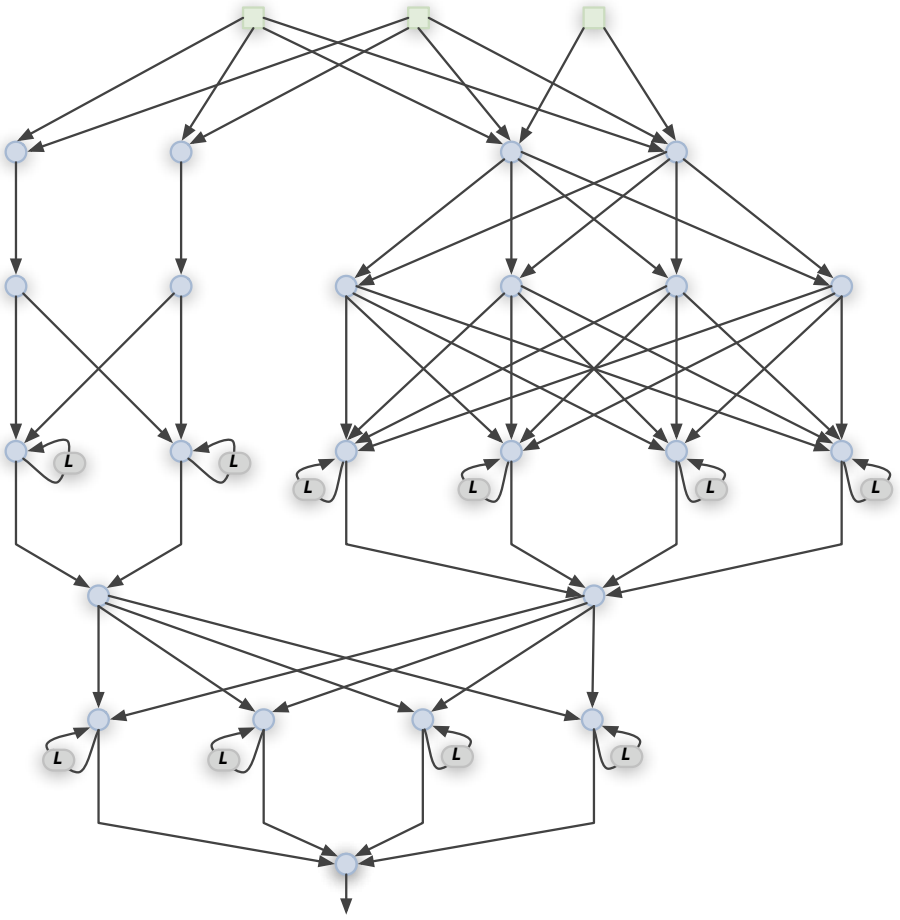
Although most of the research in machine learning has concentrated on developing individual algorithms and establishing their theoretical properties, somewhat less attention has been paid to the engineering issues that surround the *systematic composition* of learning sytems.† In practice, as illustrated above, complex learning systems are composed of a large number of individual learning components; most often, the interconnections between those components is handled in an *ad hoc* fashion by the engineer, guided by the problem at hand. This approach tends to be laborious, brittle, and hard to scale.

This chapter introduces a systematic procedure that is both correct and efficient for the creation of training sets when dealing with complex networks of processing elements, some of which are adaptive (i.e. learning algorithms), and whose performance is evaluated in a sequential validation framework, repeatedly interleaving training and testing.

The chapter is organized as follows. In section 4.2, we explain specific issues in sequential learning (particularly when dealing with economic time-series data) that make out-of-sample performance evaluation somewhat of a challenge. In section 4.3 we introduce relevant notation as well as formally defining the learning and sequential-validation framework that we shall be assuming. Section 4.4 introduces general algorithms for the creation and up-

*This would arise when the learning algorithms themselves exhibit linear time complexity with respect to the length T of the training set; since training set construction is $O(T)$, its time complexity is on the same order of magnitude as the learning part *per se*.

†These issues are quite different from those tackled by meta-algorithms such as bagging (Breiman 1996) and boosting (Freund and Schapire 1996), which are concerned with specific compositional forms to derive *new learning algorithms* with precise properties. Our concerns are somewhat related to those faced by stacking (Wolpert 1992); we discuss this point in the proceeds.

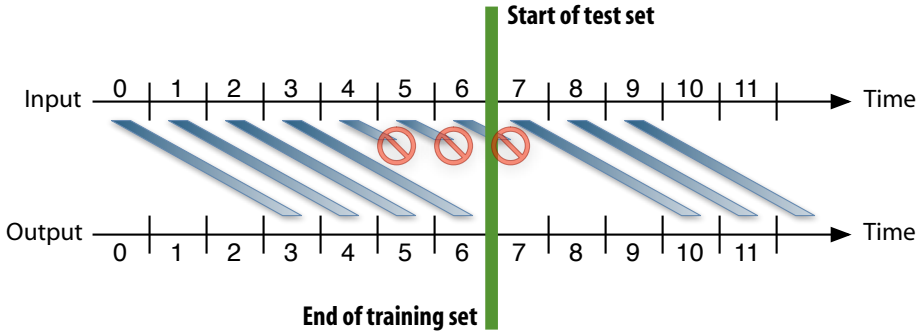


▶ **Figure 4.3.** *Example of complex composition of learning algorithms required by a financial decision-making application (outlined in §4.6/p. 114). The top row represents external inputs and each lower-level circle represents a complete learning algorithm; the arrows represent the functional data flow. The loops labeled with an L represent time-lagged connections.*

date of training sets in a temporal simulation involving learning algorithms. We continue in section 4.5 with a domain analysis aimed at contrasting alternative architectures and underscoring the necessity of the proposed approach in order to realize all desired composition scenarios. We conclude (section 4.6) with a case study of a practical deployment of the algorithms herein introduced in the context of investment fund management.

4.2 Correctness Issues in Sequential Learning

Since our main focus is on financial decision-making, we are seeking an experimental framework that is as close as possible to the situation faced by an actual economic agent who must make decisions in real time. In particular, even though typical simulations involve the use of large amounts of historical data, we rely on an experimental methodology that seeks to



◀ **Figure 4.4.** *Illustration of forecasting at horizon h; in a train–test setting, the last h observations in the training set must be discarded since their corresponding target lies "in the future" (in the test set).*

avoid *data snooping biases* (Lo and MacKinlay 1990; White 2000) to the greatest possible extent.

One simple but common cause of error when evaluating forecasting or decision models within a standard out-of-sample setting is to mistakenly use *data from the future* when constructing a model for a given day (assuming an ongoing simulation). This tends to yield an optimistically-biased estimate of future performance, since the knowledge of future outcomes obviously is of great help when trying to make a forecast of those outcomes.

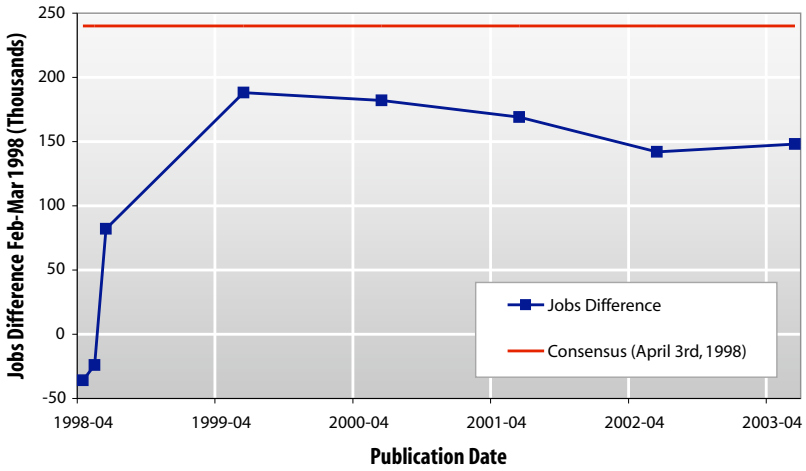
This situation is illustrated in Fig. 4.4, wherein forecasting at horizon $h = 3$ is attempted. Because of the horizon, we are capable of evaluating the quality of a decision made at time t only at time $t+3$ (on the figure). At the boundary between the training and test sets, we have h training examples for which we are never "legally" capable of measuring performance, since the target information is part of the test set. These elements must be discarded from the training set in order to arrive at a *causal* estimation of performance, for otherwise targets for the last h examples in the training set would overlap with the test set, yielding the optimistic performance estimation bias.

4.2.1 Economic Data

Causality implies that we treat macroeconomic data with particular care when performing historical simulations. Clements and Hendry (1999) note that such data series "may be inaccurate, prone to revision, and are often provided after a non-negligible delay." Contrarily to an oversimplified textbook picture where macroeconomic data series are provided and static, the reality is cloudier: most series* are updated *post hoc* through a process of successive revisions. Upon releasing a new value for most of the significant economic indicators, the past value (generally that of the previous month) is revised as well. It may happen that revisions go further in the past, as the example of Fig. 4.5 illustrates.

*At least in the United States and Canada.

It is unfortunate that most data providers only provide the **last published revision** as the "true value" of a series; this obscures the real nature



▶ **Figure 4.5.** *Example of* post hoc *revisions to macroeconomic time-series. U.S. non-farm payroll difference between February and March 1998, first published on April 4th 1998, along with the most recent consensus estimate before the release date (on April 3rd 1998). The plot shows that significant revisions are made to the data up to several years after its initial release. Data provided by the Federal Reserve Bank of St. Louis.*

of the impact of economic news releases on the financial markets. To understand this impact, it is useful to recall that most economic variables of importance are tracked by a number of analysts who make forecasts as to their outcome; the average value of these forecasts is called the *consensus estimate* for a variable. When new data is released regarding a variable, it is often argued by practitioners that the following quantities have the greatest market impact:*

- The difference between the actual value and the most recent consensus estimate (the so-called *surprise*).
- The difference between the actual value and the last available revision of the previous time-step (e.g. month) value.

Some academic studies lend credibility to this view, particularly regarding its consequences on market volatility (Hautsch and Hess 2002; Hess 2001). This "market reaction hypothesis" implies that revisions to a series value are, to a first approximation, ignored by the market.† It also suggests that carrying out simulations using the commonly-available historical macroeconomic data may introduce hard-to-quantify noise, especially when the revised values are of a different sign than the originally-published values, when compared against the consensus estimates.‡ In all cases, when one cares about the **causality of simulations**, it is incorrect to use such

*In the sense of introducing local volatility.

†To a large extent, this can be explained by the fact that when a revision is issued at a later date, its impact is submerged by that of the most recent series values. For instance, if a revision is issued in May for the March employment, its impact is overshadowed by the first release of the April employment figures.

‡It further suggests that models that use macroeconomic variables as inputs should include consensus estimates as well.

revised data: the revisions (namely, the most recent series values) would not have been available to a decision-maker acting in real time, thence should not be used.

Luckily, it is now becoming possible to have access to so-called "vintage data," containing the history of all revisions released about a series.* Availability of this data enables us to conceive of historically-accurate simulations of decision-making processes. Nevertheless, practical complications quickly come to light, especially in areas of database management and learning algorithms.

4.2.2 Vintage Data and Database Management Issues

The most obvious is related to database management issues: a "vintage series" can no longer be considered a single time series, but really is a **collection** of such series, one for the view of the past that was in effect on each given day.† One common manner of handling this data is through a *two-date* convention, wherein all time-series observations are recorded using two dates instead of one:

- The **observation date** records the nominal date for which the observation is about. For example, the unemployment rate for September 2007 would be recorded with an observation date of (say) 30 September 2007, regardless of the date at which revisions are published.
- The **publication date** records the moment at which the information becomes available. For historical time-series data, this date comes after (or precisely on) the observation date.

To give a concrete example, suppose that the unemployment rate for September 2007 is first released on 15 October 2007, then subject to two monthly revisions. The observation and publication dates would be recorded as follows in the database:

Observation Date	Publication Date
30 September 2007	15 October 2007
30 September 2007	15 November 2007
30 September 2007	15 December 2007

4.2.3 Vintage Data and Learning Algorithm Issues

The introduction of *post hoc* revisions to time-series data presents a further obstacle in the context of sequential validation: when computing, for instance, the difference between a variable and its consensus, we want this

*For instance, in the United States, the Federal Reserve Bank of St. Louis now makes vintage series available; they are what allowed us to plot Fig. 4.5 the first place.

†Although we focus on time series of historical data, the same constructions could manifestly be conceived for series of forecasts of future events.

difference to be taken using **first-published** series values—which remain constant throughout the simulation. In contrast, when using a variable as a *level* (also known as *stock variables*), we rather want to use **most-recently-published** (at the time of making the simulated decision) series values. The crucial difference is that, in the latter case, the "view of the past" regarding a series may change: this occurs when the simulation crosses a date where a new revision was published about a previous observation date.

In particular, training sets that could have been constructed incrementally in a sequential validation framework (§4.1/p. 91) may have to be mended more deeply to accommodate history revisions that could affect some series. This might require relinquishing incrementality of training-set updates in situations where it would otherwise have been feasible. Additionally, opportunities for incrementality now becomes *data-dependent*: contingent on whether and when a series is revised, training sets that depend on that series may be able to benefit from incremental updates.

Still, context-specific realities and optimizations mitigate this unpropitious outlook. In particular, because this process of history revision affects only macroeconomic series, which are generally published monthly, we maintain hopes of preserving a certain level of computational efficiency (and we assume that most reasonable simulations with financial data are at a daily or higher frequency). Furthermore, higher-frequency data (including prices) remain largely unaffected by this phenomenon.* Finally, we introduce below an optimization that efficiently deals with the scenario of revisions that are made in the last simulated time-step.

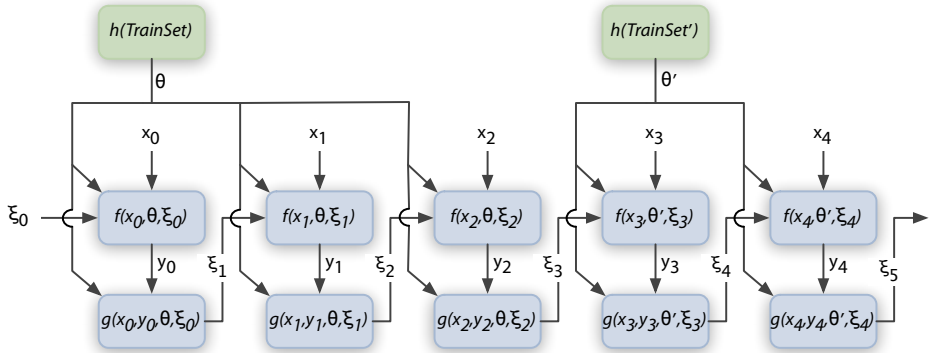
4.3 Learning Framework and Notation

We now formalize the learning concepts that we shall be using. We assume a discrete-time framework, $t \in \mathbb{N}$, where a single *time period* (e.g. a day or a month) elapses between times t and $t+1$. We further assume that T time steps of data are available, numbered from 0 to $T-1$.

4.3.1 Learning

We define learning primitives that are useful for controlling *dynamical systems*, wherein a temporal dimension is significant. Sequential validation, as introduced below, makes use of those primitives. Operationally, as depicted in Fig. 4.6, we can view the learning system as making a decision at each time-step as the result of an *output computation function* that yields an output y_t given a current input x_t, state ξ_t and parameter vector $\boldsymbol{\theta}$ (explained next). The *state* is a summary of the decisions that have been

*Although, as this is written in 2009, some providers of commodity futures data only make daily trading volume and open interest available with a 24-hour delay.



◀ **Figure 4.6.** *Illustration of the framework that is assumed for temporal learning. As outlined in the text, $f(x, \boldsymbol{\theta}, \xi)$ is an output-computation function, given current inputs x, state ξ and learned parameters $\boldsymbol{\theta}$; $g(x, y, \theta, \xi)$ is the state-transition function, and $h(\mathit{TrainSet}) \rightsquigarrow \boldsymbol{\theta}$ carries out time-independent batch training, yielding the parameter vector. Also shown is the fact that batch training can be arbitrarily interleaved with output computations and state transitions (h called with a new training set, $\mathit{TrainSet}'$).*

made in the previous time-steps and that are relevant for making the current decision. For example, the state may specify the location of an agent in a gridworld, or the currently-held assets in a portfolio management task. More generally, the state constrains the set of admissible outputs (actions) in the current time-step.* The state evolves according to a *state-transition function*, accepting the current output y_t in addition to the same inputs as the output computation function. A particular sequence $\langle x_t, y_t, \xi_t \rangle_{t=0}^{T-1}$ of inputs-outputs-states is called a *trajectory*.

Finally, the learner's behavior is assumed to be governed by a set of adjustable parameters that are contained in the parameter vector $\boldsymbol{\theta}$. These are obtained as the result of a *batch training function*, which yields $\boldsymbol{\theta}$ given a training set. Note that training, in this framework, is independent of any given trajectory and can be thought of as operating on a different level: it is both possible for the same parameters to be applied to many different trajectories (for instance, in a Monte Carlo simulation context), or for the parameters to change in the middle of a single trajectory (as illustrated in Fig. 4.6). The latter would typically occur in a sequential validation scenario.

The following definition makes those concepts more precise.

Definition 7 *A **temporal learning algorithm** is a quadruple $\langle f, g, h, \xi_0 \rangle$ where*

Output Computation *$f : \mathcal{X} \times \Theta \times \Xi \mapsto \mathcal{Y}$ is a function that given a set of inputs $\mathbf{x} \in \mathcal{X}$, a set of parameters $\boldsymbol{\theta} \in \Theta$ and a current state $\xi_t \in \Xi$ computes a set of outputs $\mathbf{y} \in \mathcal{Y}$.*†

*It is sometimes assumed that the state is a finite-dimensional vector; we shall need no such assumption about the representation of the state object in the current definition.

†The precise spaces in which these quantities lie is unimportant, but for our purposes it will be enough to assume they are *sets of named vectors of reals*, for instance $\mathcal{X} \ni x = \{\langle name_i, x_i \rangle\}$ with $name_i$ some learner-dependent identifiers and $x_i \in \mathbb{R}^{n_i}$ for some integer n_i. This representation is both more powerful and convenient than the traditional

State Transition $g : \mathcal{X} \times \mathcal{Y} \times \Theta \times \Xi \mapsto \Xi$ *is a function that given a set of inputs* $\mathbf{x} \in \mathcal{X}$*, previously-computed outputs* $\mathbf{y} \in \mathcal{Y}$*, a set of parameters* $\boldsymbol{\theta} \in \Theta$ *and a state* $\xi_t \in \Xi$ *at time* $t \geq 0$ *computes a state for the next time-step* ξ_{t+1}*. State transitions are deterministic with respect to* ξ *(which may represent a probability distribution, as we shall see below).*[*]

Training $h : \mathcal{X}^\ell \times \mathcal{Y}^\ell \mapsto \mathcal{B} \times \Theta$ *is a function that given* ℓ *input-target pairs returns a set of parameters* $\boldsymbol{\theta} \in \Theta$ *as well as an indicator* NON-TRIVIAL $\in \mathcal{B} = \{\text{TRUE}, \text{FALSE}\}$ *if the training has been "non-trivial", namely that functions* f *and* g *can be assumed to behave differently with the new set of parameters (as such, this indicator may depend on the previous set of parameters; this concept is explained more fully on p. 103).*

Initial State $\xi_0 \in \Xi$ *is an initial state from which output computation and state transition operations can be started.*

Note that we assume that an *initial training* is performed before the output computation and state transition functions can be used for the first time.

Example: The Kalman Filter

To illustrate the generality of the above framework, we briefly consider how the well-known Kalman filter ((Kalman 1960); see, e.g., Kailath, Sayed, and Hassibi (2000) for an introduction) can be expressed in terms of the above primitives. A probabilistic graphical model view of the dynamical system assumed by the filter is given in Fig. 4.7 (Roweis and Ghahramani 1999). The (uncontrolled) dynamics of the system are

$$\mathbf{u}_t = \mathbf{A}_t \mathbf{u}_{t-1} + \mathbf{w}_t, \qquad t > 0 \tag{4.1}$$

$$\mathbf{v}_t = \mathbf{H}_t \mathbf{u}_t + \mathbf{z}_t, \qquad t \geq 0 \tag{4.2}$$

with

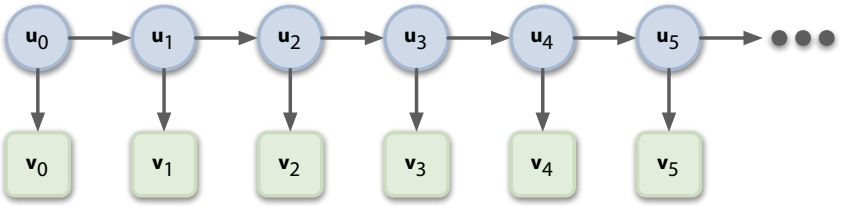
$$\mathbf{w}_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{Q}_t), \qquad \mathbf{z}_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \mathbf{R}_t), \qquad \mathbf{u}_0 \sim \mathcal{N}(\boldsymbol{\mu}_0, \boldsymbol{\Gamma}_0)$$

and $\mathbf{u}_t, \mathbf{w}_t \in \mathbb{R}^m, \mathbf{A}_t, \mathbf{Q}_t \in \mathbb{R}^{m \times m}, \mathbf{v}_t, \mathbf{z}_t \in \mathbb{R}^n$, $\mathbf{H}_t \in \mathbb{R}^{n \times m}$ and $\mathbf{R}_t \in \mathbb{R}^{n \times n}$. The notation $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ represents a normally-distributed random variable with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

---

fixed-dimensional vector provided as input to a learning algorithm, for the input to a learner can generically be specified as the set union of the outputs of the learners on which it depends, without necessitating the introduction of arbitrary concatenation rules.

[*]For notational simplicity, we assume that the state space Ξ is time-invariant, but this is not strictly required for the rest of the exposition.



◀ **Figure 4.7.** *Generative view of the Kalman filter as a graphical model.*

In this system, $\mathbf{u}_t$ represents the *state* at time t and is assumed to be *hidden* (unobserved).* It must be inferred from a sequence of observations (measurements) $\mathbf{v}_t$. From the initial conditions and system equations, and assuming known values for $\mathbf{A}_t, \mathbf{Q}_t, \mathbf{H}_t$ and $\mathbf{R}_t$, Kalman derived recursive equations for computing the distribution of the sequence of hidden states given a sequence of observations. The distribution of the initial state $\mathbf{u}_0$ is assumed to be normal with given mean $\boldsymbol{\mu}_0$ and covariance matrix $\boldsymbol{\Gamma}_0$.

We shall be concerned with the **filtering** (or tracking) view, wherein given an observation sequence that extends until current time t and a state distribution estimate for $\mathbf{u}_{t-1}$ at time $t-1$ we seek to infer the state distribution at time t.†

To express the Kalman filter in the temporal learning framework, we shall pose the following equivalences:

- The temporal-learning **set of parameters** $\boldsymbol{\theta}$ is made up of the Kalman state-transition matrices, observation matrices, and initial state distribution, respectively $\mathbf{A}$, $\mathbf{Q}$, $\mathbf{H}$, $\mathbf{R}$, $\boldsymbol{\mu}_0$ and $\boldsymbol{\Gamma}_0$. Note that although the transition and observation equations (4.1)–(4.2) support time-varying parameters, we shall assume that they do not vary in order to make learning tractable.

- The **learner state** at time t is made up of the *predicted* mean and covariance matrix of the Kalman state for time t given information until time $t-1$, respectively written as $\hat{\boldsymbol{\mu}}_{t|t-1}$ and $\hat{\boldsymbol{\Gamma}}_{t|t-1}$. Since eq. (4.1) and (4.2) are linear and the initial Kalman state $\mathbf{u}_0$ is assumed to be normally-distributed, the Kalman state for all subsequent time-steps will also be normally-distributed, and it is sufficient to keep track of

*It should be noted that "state" is used with a different meaning in discussing the Kalman filter than is used in def. 7: in the first case, the state is a single vector $\in \mathbb{R}^m$ (and follows a normal distribution under the Kalman filter assumptions); in contrast, the notion of state for def. 7 is an abstract object that encompasses arbitrary sufficient statistics about the past. In the present section, the use of "state" by itself refers to the Kalman state; when talking about state in the sense of def. 7, we shall use "learner state".

†This is different from the **smoothing view** wherein one is given *a priori* the complete observation sequence from 0 to T, from which one may infer the most likely state trajectory. However, smoothing is an acausal operation that requires knowing about the future, which we shall not further consider.

only the current mean and covariance matrix. This implies that the *learner state* consists of just these two quantities.

- The temporal-learning **input** is an observation vector $\mathbf{v}_t$ provided to the Kalman filter. Note that what is termed "input" here is really an "output" of the generative model (4.2); however, from the learner standpoint, these observations are the inputs required for inference. The Kalman filter can also be augmented to cover the *controlled case*, wherein an exogenous control variable is used to alter the state dynamics (4.1); in this case, the control variable would also be provided as part of the learner inputs.

- The temporal-learning **output** is a *corrected* mean and covariance matrix of the Kalman state, that incorporates the information given by the observation vector $\mathbf{v}_t$. These are written as $\hat{\boldsymbol{\mu}}_{t|t}$ and $\hat{\boldsymbol{\Gamma}}_{t|t}$.

From these equivalences, the temporal-learning operations are defined as follows:

Initial State Consists of $\boldsymbol{\mu}_0$ and $\boldsymbol{\Gamma}_0$ obtained by a previous call to the training procedure.

Output Computation Correct the state $\hat{\boldsymbol{\mu}}_{t|t-1}$ and $\hat{\boldsymbol{\Gamma}}_{t|t-1}$ to take into account the time-t observation $\mathbf{v}_t$. Start by computing the residual between the observation and its expected value under the predictive distribution

$$\hat{\mathbf{z}}_t = \mathbf{v}_t - \mathbf{H}\hat{\boldsymbol{\mu}}_{t|t-1}$$

whose covariance matrix is given by

$$\mathbf{S}_t = \mathbf{H}\hat{\boldsymbol{\Gamma}}_{t|t-1}\mathbf{H}' + \mathbf{R}$$

yielding the so-called *Kalman gain* factor

$$\mathbf{K}_t = \hat{\boldsymbol{\Gamma}}_{t|t-1}\mathbf{H}\mathbf{S}_t^{-1}.$$

The updated estimators for the state mean and covariance matrix are seen as *corrections* to the previous estimators, and given by

$$\hat{\boldsymbol{\mu}}_{t|t} = \hat{\boldsymbol{\mu}}_{t|t-1} + \mathbf{K}_t\hat{\mathbf{z}}_t$$

and

$$\hat{\boldsymbol{\Gamma}}_{t|t} = (I - \mathbf{K}_t\mathbf{H})\hat{\boldsymbol{\Gamma}}_{t|t-1},$$

with I the identity matrix. The last two quantities constitute the output of the learner at time t.

State Transition Compute the next state as a linear forecast implied from the state-transition equation (4.1), the result of the current output (corrected mean $\hat{\boldsymbol{\mu}}_{t|t}$ and covariance matrix $\hat{\boldsymbol{\Gamma}}_{t|t}$),

$$\hat{\boldsymbol{\mu}}_{t+1|t} = \mathbf{A}\hat{\boldsymbol{\mu}}_{t|t} \tag{4.3}$$

$$\hat{\boldsymbol{\Gamma}}_{t+1|t} = \mathbf{A}\boldsymbol{\Gamma}_{t|t}\mathbf{A}' + \mathbf{Q} \tag{4.4}$$

Training Model parameters $\mathbf{A}$, $\mathbf{Q}$, $\mathbf{H}$, $\mathbf{R}$, $\boldsymbol{\mu}_0$ and $\boldsymbol{\Gamma}_0$ can be estimated by the Expectation-Maximization algorithm (Dempster, Laird, and Rubin 1977; Neal and Hinton 1998) to maximize the likelihood of sequences of observations within a training set. Derivations of the EM reestimation equations specific to the Kalman filter are given by Shumway and Stoffer (1982) and Ghahramani and Hinton (1996).

Additional Remarks

Some additional remarks are in order regarding the definition of a temporal learning algorithm:

▶ **Training versus State Variables :** Training does not involve any state variable; in other words, training, as defined above, occurs in "batch mode", independently of any current test trajectory. Philosophically, we can consider that training defines the identity of a learner (in general), whereas states govern its behavior in a particular context. The prototypical example of such a learner is the previously-introduced Kalman Filter, wherein transition matrices are estimated by maximum likelihood from a large historical dataset during training, but filtering for a particular trajectory is performed by the combination of output-computation and state-transition functions.*

▶ **Pure Preprocessing or Online Learning :** We allow Θ to be the empty set $\emptyset$. This is useful to allow *non-adaptive* elements in a temporal learning network, namely fixed processing elements or strictly online learning modules (Saad 1999), for which all learned parameters occur as state variables. This special case is recognized and optimized for by the algorithms introduced in §4.4/p. 105.

▶ **Input Data Representation :** Raw data can be represented as learners that do not take any inputs,† for which $\Theta = \emptyset$, and that output "constant" results that depend only on the current time step. History revisions can be

*Also note that the separation between training and output computation/state transition implies that the *interpretation* of the state variables is independent from the results of training.

†Besides the current and previous "end-of-training" dates, which can be viewed as meta-inputs. As explained below, this lets the learner know that a history revision boundary has been crossed in the sequential validation.

effected by returning the indicator NON-TRIVIAL = TRUE from the training function h when such revisions occur; this notifies dependent learners that incrementality assumptions no longer hold on their training sets. Note that in the current framework, history revisions are relevant only during training time, which operates on the entire data history. (In contrast, output computation time operates only on the incremental information available for the new time-step.)

▶ **Bayesian Estimation :** The training function h can also encompass the Bayesian case, wherein one does not estimate a single vector of parameters, but a *posterior probability distribution* (given the training data) over the space of parameters. This amounts to having a $\boldsymbol{\theta}$ with a more complex structure, which our notation does not prohibit. We would also assume that the prior distribution over parameters is passed along with the training set. However, for simplicity, the rest of this chapter assumes the point-estimation case.

4.3.2 Temporal Learning Networks

As stressed in the introduction, typical learning algorithms do not occur in isolation; we now define the graph structure that enables their composition.

Definition 8 *A **temporal learning network** is a directed graph $G = \langle \mathcal{V}, \mathcal{E} \rangle$ where*

Vertices $\mathcal{V} = \{v^j\}$ *is a set of J temporal learners (elements that satisfy def. 7) $v^j = \langle f^j, g^j, h^j, \xi_0^j \rangle, j = 1, \ldots, J$;*

Edges $\mathcal{E}$ *is a set of triples $\langle i, j, k \rangle$, where $v^i, v^j \in \mathcal{V}$ represent a directed link between learners v^i (source) and v^j (destination), and $k \in \mathbb{N}$ is a temporal lag on the connection, as explained below.*

Furthermore, we let $\mathcal{E}_0 \stackrel{\triangle}{=} \{\langle u, v, k \rangle \in \mathcal{E} : k = 0\}$, the set of lag-0 edges. The subgraph $G_0 = \langle \mathcal{V}, \mathcal{E}_0 \rangle$ must be acyclic. We assume, without loss of generality, that the learners are numbered to satisfy the topological ordering, such that $\langle i, j, 0 \rangle \in \mathcal{E}_0 \Longrightarrow i < j$.

The intent of this definition is to capture output–input connections between learners. A lag-0 edge $\langle u, v, 0 \rangle$ represents the basic building block of functional composition: we assume that the output computed by learner u at time t serves as input to learner v at the same time-step.* Likewise, a lag-k edge $\langle u, v, k \rangle, k > 0$ represents a delayed connection: the output of u at time $t - k$ is used as the input of v at time t. Clearly, in order to have well-defined propagation at time t, we require the subgraph of lag-0 edges

*If learner v receives multiple inputs, it is provided with the union of inputs coming from all sources.

$\mathcal{E}_0$ to be acyclic such that an ordering exists in which to perform the output-computation operations; this can be found by an elementary topological sort algorithm (Cormen, Leiserson, Rivest, and Stein 2001). No such ordering is required for higher-order lags. We will assume that a sentinel "missing value" $\perp$ is returned whenever a value for a time-step $t < 0$ is required.

4.4 Algorithms

In this section, we present a set of algorithms to evaluate the performance of a temporal learning network within the sequential validation framework, as well as efficiently updating the training set of each learner.

The algorithms are introduced in several logical parts. First, the overall driver of the simulation is the SequentialValidation procedure, which does not present any technical difficulty. We then present the training-related procedures (Train and UseOnTrain) and the output-computation and state-transition ones (ComputeOutput and TransitionState). The latter procedures act as "wrappers" around the primitive functions f^j, g^j and h^j for each learner j, introduced in §4.3.1/p. 98. Moreover, ξ_0^j is assumed to denote the initial state of learning j, and the algorithms operate on a given temporal learning network $\langle \mathcal{V}, \mathcal{E} \rangle$. For simplicity of exposition, we assume a strict sequential validation framework, in particular that we have a single time history which is used for creating both the training and test sets.

4.4.1 Variables and Notation

Table 4.1 lists the global variables that are assumed to persist between procedure calls. In a concrete implementation, it is straightforward to convert those to an "object-oriented" arrangement.

Regarding notation, we assume we can take a time-t "slice" of a training set X^j by writing X_t^j, which produces an element suitable for using as input to the output-computation function f^j. We further assume that we can row-wise construct such a training set by assigning to each time-t slice separately, e.g. $Y_i \longleftarrow f(\ldots)$.

4.4.2 Sequential Validation

The SequentialValidation procedure starts (lines 3 and 4) by initializing the state variables corresponding to a **train** and a **test** trajectory. The need for two separate trajectories arises from the necessity to call the output-computation functions f^j from two separate contexts: first, during the normal test step of sequential validation (see Fig. 4.1) to compute outputs on *test inputs*, and second after training learner j to compute the outputs on *train inputs* (i.e. on the elements of the training sets), which is necessary for constructing the training set of any dependent learner. As will be made clear

Table 4.1. *Global variables assumed to persist between procedure calls*

Variable	Procedure(s)	Meaning
$\hat{\xi}_t^j$	SequentialValidation, ComputeOutput, TransitionState	Learner j state variable at time t for the test trajectory
$\tilde{\xi}_t^j$	SequentialValidation, Train, UseOnTrain	Learner j state variable at time t for the train trajectory (only one such trajectory is required)
$\boldsymbol{\theta}^j$	Train, UseOnTrain, ComputeOutput, TransitionState	Learner j current parameters
X^j	Train, UseOnTrain	Learner j current training set (contains both inputs and targets)
Y^j	Train, UseOnTrain	Learner j outputs computed on the current training set
x_t^j	ComputeOutput, TransitionState	Learner j test-time input variables at time t
y_t^j	ComputeOutput, TransitionState	Learner j test-time output variables at time t, for $t \geq 0$; "missing value" $\perp$ for $t < 0$

below in the description of Train and UseOnTrain, it is frequently necessary to reinitialize the training trajectory to time-step 0, but the test trajectory should never be reinitialized during the execution of sequential validation, since this would correspond to rewriting history (and would be an action unavailable to a decision-maker).

Within the main loop of SequentialValidation (lines 8–13), output computation (testing) occurs at every time-step while training is carried out according to a completely-independent, asynchronous, schedule passed as an argument to the procedure. The *training-schedule* is simply the set of time-steps on which training is allowed to occur, where we assume that $0 \in$ *training-schedule*, so that outputs are never computed on an untrained learner.* At time t, the outputs are always computed given the most recent parameter estimates, resulting from training at time $t_{previous}$.†

Listing 4.1: Sequential Validation

```
1 def SequentialValidation(training-schedule):
2     # Initialize test (ξ̂) and train (ξ̃) trajectories
3     ξ̂_0^j ⟵ ξ_0^j,   j = 1, ..., J
```

*This is more a matter of notational convenience: absent an obvious default behavior, learners trained on too little data or "no data" can be defined to output missing values or other sentinel.

†With this arrangement, care must be taken when the learner parameters can be used to introduce a scale transformation on some variables, which in turn are used as *lagged inputs* to other learners. For example, suppose that learner 1 computes the principal components (PCA) of a set of input variables, whose first-differences $\text{PCA}_t - \text{PCA}_{t-1}$ are used as inputs by learner 2. Assume that a retraining occurs at time τ. If lagging is implemented "naïvely" by buffering the previous-time outputs, at time τ the PCA difference would use PCA_t computed with time-τ parameters, but PCA_{t-1} computed in an altogether different space with time-$\tilde{\tau}$ parameters, where $\tilde{\tau}$ is the time of the previous training.

```
4       ξ̃_0^j ⟵ ξ_0^j,   j = 1, ..., J
5       t_previous ⟵ −1
6
7       # Interleave training and testing
8       for t in 0, ..., T − 1:
9           if t ∈ training-schedule:
10              Train(t, t_previous)
11              t_previous ⟵ t
12          ComputeOutput(t)
13          TransitionState(t)
```

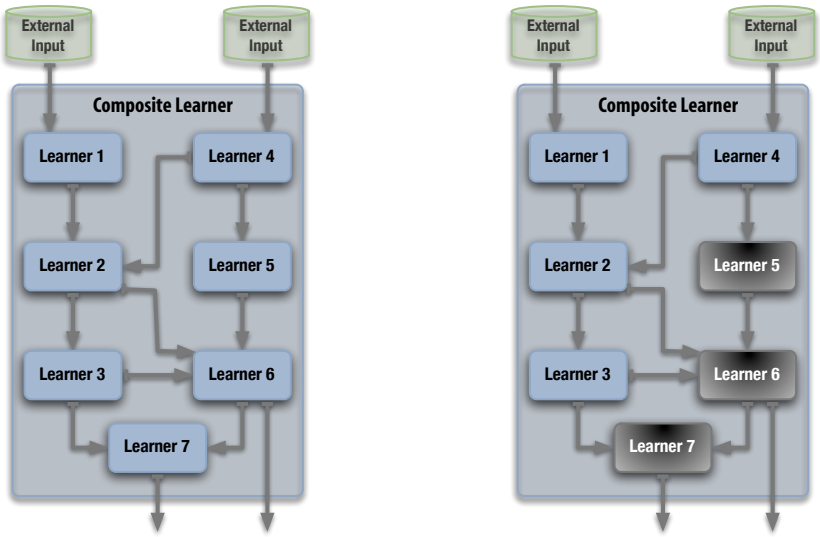
4.4.3 Training

The Train procedure wraps individual learners' training functions h^j and keeps track of a "dirtiness" status *is-dirty*^{j} for each learner. This status determines whether the associated UseOnTrain procedure is allowed to compute the learner's outputs on the training set incrementally or not. Train starts by considering each learner "clean" (lines 2–3). Then, in the order given by the topological sort of the lag-0 edges in $\mathcal{E}$, it constructs the training set for learner j from the outputs on the training set of learners connected to j through a lag-0 edge (line 7). For simplicity of notation, this Train pseudocode does not handle delayed connections at the training level. If training sets can be viewed as matrices, then this is easily handled through shifting the inputs matrices Y^i by an appropriate number of rows and introducing rows of missing values as appropriate.

The results of the learner's training function h^j are used in two ways: First, the indicator of whether the training was non-trivial is used to mark the learner dirty if it was not already; a "dirty learner" then propagates its filth to its train-time dependents (lines 13–15). This process is illustrated in Fig. 4.8. Finally, the new parameters $\boldsymbol{\theta}^j$ are used to compute the learner's outputs on the training set through UseOnTrain.

Listing 4.2: Training

```
1   def Train(t, t_previous):
2       for j in 1, ..., J:
3           is-dirty^j ⟵ FALSE # Initially, everybody is clean
4
5       for j in 1, ..., J:
6           # Create training set X^j for learner j
7           X^j ⟵ ⋃_{⟨i,j,0⟩∈ℰ} Y^i
8           # Carry out the training per se
9           ⟨non-trivial^j, θ^j⟩ ⟵ h^j(X^j)
10          # Dirty−up descendants if learner j is dirty
11          if non-trivial^j:
12              is-dirty^j ⟵ TRUE
```



▶ **Figure 4.8.** ***Left:*** *Example of a directed acyclic graph of composed learning modules.* ***Right:*** *Illustration of how the "dirty" status of a learner propagates to its descendants during training. As soon as a learner (here Learner 5) exhibits non-trivial retraining or its historical input data underwent revisions, its outputs on the training set must be recomputed anew, indicated by the darker shade. This applies recursively to all learners that directly or indirectly depend upon it.*

```
    if is-dirty^j:
        for ⟨j, j̃, 0⟩ ∈ ℰ:
            is-dirty^j̃ ⟵ TRUE
    # Finally compute outputs on training−set inputs
    UseOnTrain(j, t, t_previous)
```

The UseOnTrain procedure is the heart of the compositional step. It allows the training set of a learner to be created from the results of training a predecessor j, by computing the outputs of learner j on each row of its training set. Two scenarios need to be considered:

1. If the learner is marked "dirty" (resulting from non-trivial training or changes to the training set), the train-time state variables $\tilde{\xi}_0^j$ need to be reinitialized (line 4) in order to start computing the outputs from the first element of the training set. Then a sequence of output computations (f) and state transitions (g) are performed for each element of the training set, until the last time-step t (lines 5–7).

2. If, on the other hand, the learner has remained "non-dirty" after its previous training, the training outputs can be computed incrementally from the previous results of UseOnTrain, where the starting time-step $t_{previous}$ is passed as an argument. This loop is very similar to the previous one, but does not involve reinitializing the train-time state variables (lines 9–11).

Listing 4.3: Training Output Computation

```
1  def UseOnTrain(j, t, t_previous):
2      # If learner is dirty, start a new trajectory
```

```
    if is-dirty^j:
        ξ̃^j_0 ⟵ ξ^j_0
        for τ in 0, ..., t:
            Y^j_τ ⟵ f^j(X^j_τ, θ^j, ξ̃^j_τ)
            ξ̃^j_{τ+1} ⟵ g^j(X^j_τ, Y^j_τ, θ^j, ξ̃^j_τ)
    else:
        for τ in t_previous + 1, ..., t:
            Y^j_τ ⟵ f^j(X^j_τ, θ^j, ξ̃^j_τ)
            ξ̃^j_{τ+1} ⟵ g^j(X^j_τ, Y^j_τ, θ^j, ξ̃^j_τ)
```

4.4.4 Output Computation

The ComputeOutput procedure traverses each learner j in the order given by the topological sort of the lag-0 edges in $\mathcal{E}$. For each, it constructs the inputs x_t^j for the learner from the outputs $y_{t'}^i$ of its "predecessors" in $\mathcal{E}$, whose existence the topological sort guarantees for lag-0 dependencies, and the top-level sequential validation loop likewise guarantees for lag-k dependencies, $k > 0$. As mentioned previously, we assume that an output y_t^i for $t < 0$ maps to a missing value default.

Listing 4.4: Output Computation

```
def ComputeOutput(t):
    for j in 1, ..., J:
        # Construct the input x^j_t for learner j
        x^j_t ⟵ ⋃_{⟨i,j,τ⟩∈E} y^j_{t−τ}
        # Compute output for learner j given input x^j_t
        y^j_t ⟵ f^j(x^j_t, θ^j, ξ̂^j_t)
```

Finally, the TransitionState procedure steps the state variables of each learner forward, given the current inputs and outputs.

Listing 4.5: State Transition

```
def TransitionState(t):
    for j in 1, ..., J:
        ξ̂^j_{t+1} ⟵ g^j(x^j_t, y^j_t, θ^j, ξ̂^j_t)
```

4.4.5 Refinements and Limitations

At the expense of some increased complexity, the practical efficiency of these algorithms can be improved (albeit not their asymptotic time complexity). In particular, the UseOnTrain procedure may benefit from the following improvement: a more sophisticated specification of the training function h would report on not only whether training has been non-trivial, but also

from *where in the past* the newly-trained learner would want to revise its outputs. If a complete history of the training state $\tilde{\xi}_\tau^j$ is kept along the training set, this allows us to avoid resetting the state to zero when a "non-trivial training" has occurred (line 4 of UseOnTrain); instead we can only go back to the time of the furthest revision.

A lesser benefit can be had at no space cost in the case of history revisions occurring at the last time-step of a given training set. It comes from the recognition that the very last state-transition in the UseOnTrain loops of lines 5–7 and 9–11 (for $\tau = t$) can be delayed until the start of the next UseOnTrain call without changing behavior. (A casual perusal of this function reveals this transformation leaves unchanged the overall ordering of the calls to the underlying f^j and g^j functions.) Delaying this state-transition opens up the option of *entirely omitting the transition* should it be found unnecessary. Consider, within a current call to UseOnTrain, a history revision that is anticipated to occur in the very last time-step of the training set (currently at time t, which would become time t_{previous} in the next call to UseOnTrain). If such a revision indeed occurs, UseOnTrain can handle it by starting the loop of line 9 at time t_{previous} (instead of $t_{\mathit{previous}} + 1$), *overwriting the previously-computed outputs with those computed from the revised inputs.* If no revision occurs, UseOnTrain simply computes the missing state-transition for $\tau = \mathit{previous}$ before starting loop 9–11.

Although this special case seems rather far-fetched, we remark that it corresponds to the real-world situation of a simulation with monthly data, with the common case of macro-economic series subject to last-month revisions.

We briefly mention two further obvious improvements: since UseOnTrain is only required to be called on a learner to prepare the training set for its children in $\mathcal{E}$, this call is evidently not required for leaf learners, and may be omitted. In addition, the overall partitioning of the learning problem along a graph structure naturally lends itself to a parallel implementation for regions of the graph that show no dependencies.

Finally, one must consider an evident limitation of the approach: the memory cost of storing the individual training sets and training outputs for each learner. This cost scales in the expected way, but may become large when many learners and/or long sequences are involved. This should be weighted against the computational efficiencies of not to having to update all training sets before each retraining, a classic time-versus-space trade-off.

4.5 Domain Analysis

The definition of temporal learning algorithm introduced in §4.3.1/p. 98 lends itself well to composition through network structures. In this section we present a short domain analysis to better understand why these

structures are necessary over the arguably simpler hierarchical (tree-like) alternatives for a number of useful machine learning problems.

4.5.1 Hierarchical Designs

In a hierarchical composition of "learning modules", a single learner owns a set of "sublearners" which can be used for performing auxiliary work. The *training* and *output computation* functions of a learner are trivially defined as first recursively calling the corresponding function of each sublearner, and then carrying out any required additional work. As to the state representation in the case of dynamic models, two possibilities arise:

- Each learner can *contain* its own state*, which is directly controlled (e.g. reset to an initial starting point) by an interface provided by the learner. However, this type of interface, by almost completely abstracting away the state representation, makes it unduly difficult to operate the learner along multiple parallel trajectories. In a compositional context, this is always necessary in order to compute the fitted values on the training set in order to construct the input needed by a subsequent learner. In a simulation context, Monte Carlo trials require a large number of separate states.

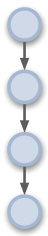
- Each learner can be passed a state vector at output-computation time; this vector contains both the state required by the learner itself, but also the concatenation of state vectors required by sublearners. This arrangement poses two difficulties: first it requires the state of the sublearners to be of a known fixed size; second, and this is a general problem with hierarchical composition, it does not allow the work done by a learner to be shared by others. This second situation is detailed next.

*A **has-a** relationship in object-oriented parlance; see, e.g., Booch et al. (2007).

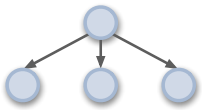
4.5.2 Network Designs

A general problem with a hierarchical decomposition of learners is that it is impossible to factor out computations that would be required by two separate learners. This operation requires a more general graph representation. Directed acyclic graphs are sufficient for a large number of causal time-series forecasting and temporal decision-making tasks, as the following cases illustrate:

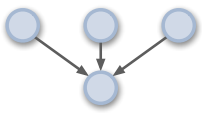
▶ **Chain Graphs :** Chain graphs correspond to a normal processing pipeline, starting with input variable preprocessing (including, for instance, dimensionality reduction), followed by a traditional learning stage, and postprocessing. They also encompass the case of *stacking* (Wolpert 1992), wherein a cascade of learners correct previous ones' errors.



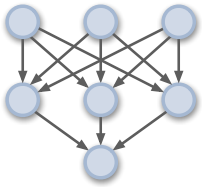
▶ **Fan Out :** This corresponds to factored-out computations, such as the use of a common covariance matrix in otherwise-parallel tasks.



▶ **Fan In :** This corresponds to the combination of several previously-computed results; in machine learning, a classic example is the bagging algorithm of Breiman (1996). In a financial setting, one can view the mean-variance optimization step of a portfolio selection problem (Markowitz 1959) as a graph of this type, wherein expected returns and covariances are first estimated (along with additional inputs such as market equilibrium weights and investment views if one is following a more robust methodology such as Black-Litterman allocation (Black and Litterman 1992; Fabozzi, Kolm, Pachamanova, and Focardi 2007)), followed by their combination in the optimization step *per se*.



▶ **Lattices :** More complex lattices arise naturally in the context of *hyper-mixtures*. These are a generalization of mixture models that occur when one is performing sequential hyperparameter optimization. Consider an exponentiated gradient mixture (Herbster and Warmuth 1998; financial applications are considered in Helmbold, Schapire, Singer, and Warmuth 1998) that combines, through weighted averaging, the outputs of a set of base models (which are represented as the top layer in the illustration). At time-step t, the mixture associates a weight $w_{t,i}$ to model i. The weights are adapted at each time-step according to an error criterion aiming at putting more importance on the recently better-performing models. This adaptation is subject to a set of hyperparameters Θ controlling, for instance, the adaptation speed and the maximum weight that can be put on a single model. The problem is that one does not know *a priori* what are good settings for these hyperparameters. In the spirit of sequential validation, a solution is to run a number of such mixtures in parallel (depicted as the middle layer in the illustration), each with its own hyperparameters Θ_j, and have the hyper-mixture (bottom node in the illustration) select the best one on the basis of the past performance of each mixture.*



In order to avoid repeated computation, it is desirable to let the base models be *shared* among the mixtures, each one then performing its own independent weighting. At time t, this yields the following sequence of computations:

*One may argue that this does not completely solve the problem since the time horizon over which the performance is evaluated at the hyper-mixture level is itself an hyperparameter; in practice, this hyperparameter is much easier to choose "reasonably" according to the characteristics of the problem than those at the mixture level.

1. The set of base models $\{i\}$ compute their outputs.

2. The individual mixtures, each with adaptation hyperparameters Θ_j, combine these outputs according to weights $w^j_{t,i}$; the mixtures independently update their weights.

3. The hyper-mixture selects the best mixture at time t according to the recent performance of each mixture.

It should be emphasized that time-delays (i.e. using as input a result that was computed in a previous time-step) do not introduce an ordering dependency in the directed acyclic graph; they simply correspond to buffering operations that must be carried out during propagation.

State Representation

In §4.5.1/p. 111 we examined why state containment and simple aggregation are inappropriate for a variety of sensible use-cases. The solution to state representation is to designate a *state-holder object* which associates a learner reference to an arbitrary state for the learner; each learner uses the state-holder object to find its own state. It is the state-holder object, and not individual state objects, that is communicated among learners within the compositional steps of §4.4/p. 105.

4.5.3 Why Separate Training and Output Computation?

A sequential learning setting is often associated with *online learning* (Saad 1999), in which model parameters are adapted contemporaneously with the computation of outputs, as examples are presented to the learner. As such, it can be asked why the framework introduced in §4.3/p. 98 distinguishes between training and output computation. Our prototype example for answering this question is to consider the Kalman Filter considered in §4.3.1/p. 100, which has both parameters and dynamic state. The output function determines the model output given its state, and the state-transition function determines the next state from the current one. The training procedure corresponds to the estimation of parameters necessary for both the output and state-transition functions.

In this framework, training operations correspond to *batch training*: given a complete past history, we can use it all (without cheating or necessarily taking temporality into account) to update the model parameters. These should be considered offline (time-invariant) parameters, independent of the current state. Furthermore, the training set needs not correspond to a single time history; several histories (either generated by a random process, or corresponding to several parallel histories of real data, e.g. many historical stock price histories) may be combined into a single training set, with the resulting maximum likelihood estimation remaining well defined.

In contrast, in the traditional online learning context, parameters are updated at each time-step within a trajectory as a new example is received. This is easily accommodated in the framework of §4.3/p. 98 by considering the (online) parameters to be part of the state vector. In this context the state transition equation may be viewed as forming the learning step. Batch training is simply omitted in this case.

4.6 Case Study

In this final section, we review the implementation of a complex financial decision-making system making use of the composition primitives introduced previously. This system is designed to automate the trading a portfolio of commodity futures (see Hull (2005) for an introduction), and relies on a number of learning algorithms at various points in the process of turning raw data into actionable portfolio recommendations.

4.6.1 Existing Systems

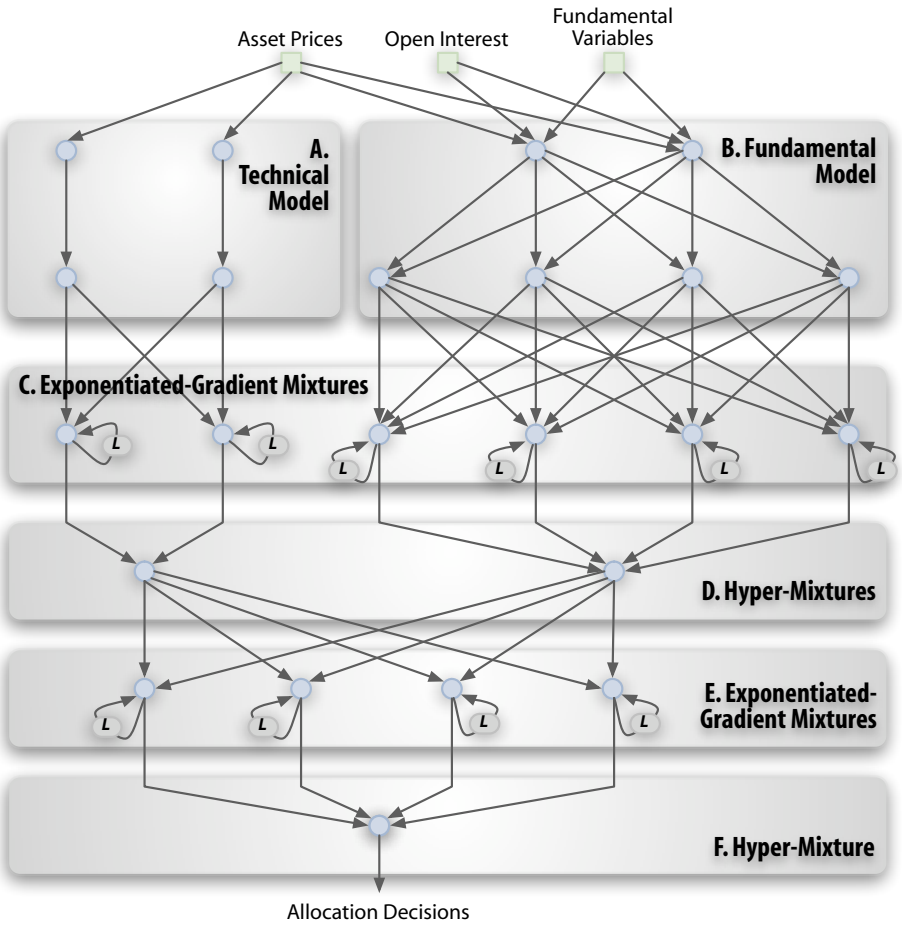
The majority of Commodity Trading Advisors (CTAs) and automated trading systems use active methodologies, of which trend following in one fashion or another is especially prevalent (Fung and Hsieh 2001; Spurgin 1999); these methods rely on the postulate that past price movements are good predictors of future ones. This is in part motivated by theoretical findings that document the evidence of abnormal returns from momentum strategies in commodities after adjusting for systematic and time-varying risks (Erb and Harvey 2006; Miffre and Rallis 2007), echoing the classical results of Jegadeesh and Titman (1993) for equities.

Unfortunately, many of the systems deployed in practice suffer from *retrospective bias* to some extent, namely that the parameters governing the trading rules are adjusted *ex post* to yield good past performance, without any guarantees other than a trader's "gut feeling" that they would perform acceptably in the future.

The design of a "cheating-free" trading system can only be realized by reproducing the real-time actions of a decision-maker, making sequential validation a natural framework. This system has been developed and used in an industrial context.

4.6.2 Making Use of the Composition Primitives

Figure 4.9 illustrate the overall system, decomposed into relevant functional blocks. We see that it consists of two "core models" (at the top), a technical model mostly making use of recent asset price variations, and a fundamental model, which can exploit more complex relationships between assets and is used to construct a mean-variance efficient portfolio. The rest of the system

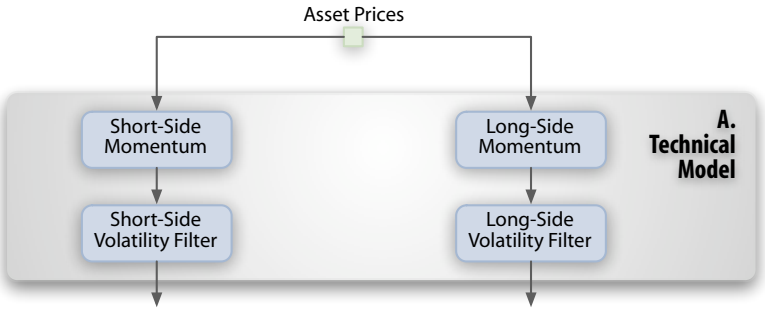


◀ **Figure 4.9.** *Functional blocks of a financial portfolio allocation system making use of the composition primitives introduced previously. The square (green) vertices represent external inputs. The round (blue) vertices represent temporal learning algorithms in the sense of definition 7. The loops labeled with an L represent time-lagged connections.*

(various mixture models) is used to combine the two basic building blocks. As will be made clear below, the apparent complexity of this combination arises because we want to set as few hyperparameters as possible from the outside and let the system adaptively choose the best hyperparameter settings.

▶ **A. Technical Model :** The first "core" building-block, illustrated in Fig. 4.10, is a traditional moving-average based momentum model that takes long and short positions based on an assetwise comparison between the current price and the one-year price moving average.* We introduce a distinction between portfolios of long and short positions, based on the observation that momentum tends to be more persistent for shorts than longs (Miffre and Rallis 2007). More specifically, let p_{it} the price of asset i at time t

*"Long" means a buying position in the asset, "short" means a selling position, and "neutral" means that no position is taken. Commodity futures can as easily be sold short as they can be bought, without the restrictions that affect equities.



▶ **Figure 4.10.** *Details of the technical model.*

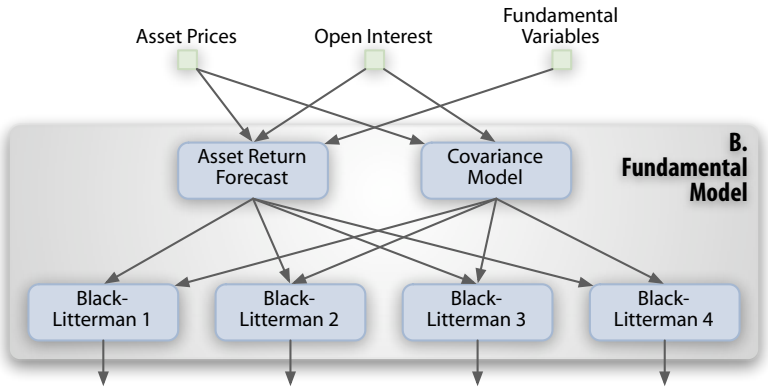
and $\hat{p}_{it}$ the one-year moving average of the same quantity. The "long-side momentum" rule is simply to recommend a long position if $p_{it} > \hat{p}_{it}$ and a neutral position otherwise; conversely, the "short-side momentum" rule recommends a short position if $p_{it} < \hat{p}_{it}$, and a neutral position otherwise. After the momentum recommendation is made, it is filtered according to the volatility rule of Dunis and Miao (2006); this rule sets to neutral the momentum model positions in periods of high market volatility. In terms of the learning primitives of §4.3.1/p. 98, each learner in the technical model consists of fixed decision rules with online estimation of key thresholds (current price moving average and volatility). As such, they do not make use of the batch training facilities.

It remains to combine the resulting independent long and short portfolios; this is performed adaptively using a mixture described below.

▶ **B. Fundamental Model :** The fundamental model seeks to exploit deeper relationships between assets, trying to estimate asset expected returns and covariances over the next month and constructing a mean-variance efficient portfolio (Markowitz 1959) to exploit the relationships that have been found. Unusual for commodity trading systems, we rely on the Black and Litterman (1992) methodology of portfolio optimization as a second building-block (see Fig. 4.11). This makes use of an asset expected-return predictor, which uses locally-weighted linear ridge regression (Hoerl and Kennard 1970) from a number of technical and macroeconomic input variables, with hyperparameters—such as the ridge coefficient and the number of neighbors in the local window—optimized by a grid search at each time-step over a validation set.

We also require an asset-return covariance matrix estimator, an exponentially-weighted moving average following the RiskMetrics (1996) methodology being used for simplicity. This carries out a recursive (online) update of the current covariance estimator $\hat{\mathbf{\Sigma}}_t$ given the previous estimator and the vector of time-t asset returns $\mathbf{r}_t$,

$$\hat{\mathbf{\Sigma}}_t = \lambda\, \hat{\mathbf{\Sigma}}_{t-1} + (1-\lambda)\mathbf{r}_t\mathbf{r}_t',$$



◀ **Figure 4.11.** *Details of the fundamental model.*

where the decay factor $\lambda = 0.94$ was used, consistently with the Riskmetrics recommendations for daily data.

The Black-Litterman model also requires the market capitalization of each asset in order to determine the asset equilibrium returns, which are part of the procedure. Since commodities do not have a capitalization in the traditional sense, we relied on the value of the outstanding open interest as a capitalization proxy.

Black-Litterman (B-L) finds the mean-variance-efficient portfolio weights, namely the result of the optimization problem

$$\mathbf{w}_t^* = \arg\max_{\mathbf{w}} \mathbf{w}'\boldsymbol{\mu}_t^{BL} - \frac{\gamma}{2}\mathbf{w}'\hat{\boldsymbol{\Sigma}}_t\mathbf{w},$$

where $\boldsymbol{\mu}_t^{BL}$ is the vector of expected asset returns obtained by the Black-Litterman estimator (which uses the raw asset return estimator; see Black and Litterman (1992) for details), and γ is the investor risk-aversion coefficient. The B-L allocation model relies on a number of such hyperparameters. Since we do not know *a priori* what should be good values for them, we run a number of B-L allocations in parallel (denoted "Black-Litterman $1 \ldots 4$" on the figure), and mix them according to a mixture.

In terms of learning primitives, of the components depicted in Fig. 4.11, the asset-return forecast contains a full-scale training operation and does not need to carry trajectory-specific state. In contrast, the covariance model uses online estimation, and hence can dispense with the necessity for batch training. The Black-Litterman model neither needs training nor specific state: the portfolio optimization is performed on-the-fly within the output-computation operation.

▶ **C. Exponentiated-Gradient Mixtures :** Despite their differences, a common thread to both the technical and fundamental models is that they *output several parallel hypotheses*, corresponding to different choices of hyperparameters (the mixing proportions of the long and short models in

the cast of the technical model, and the B-L hyperparameters in the case of the fundamental model). Instead of carrying out hard model selection, we combine these hypotheses with the help of a mixture, using the fixed-share version (Herbster and Warmuth 1998) of the exponentiated gradient algorithm (Kivinen and Warmuth 1997). This method uses a multiplicative update of the weights, followed by a redistribution step that prevents any of the weights from becoming too large. Briefly, this mixture operates as follows. Let $C_{t,k}$ a performance measure observed for model k (out of a total of K models) during period t. The weight given to each model is updated according to:

1. *Performance Update:*

$$w^m_{t,k} = w^s_{t,k} \exp(\eta\, C_{t,k}),$$

2. *Fixed Share:*

$$\begin{aligned} \text{pool}_t &= \alpha \sum_{k=1}^{K} w^m_{t,k} \\ w^s_{t+1,k} &= (1-\alpha) w^m_{t,k} + \frac{\text{pool}_t - \alpha w^m_{t,k}}{K-1}, \end{aligned}$$

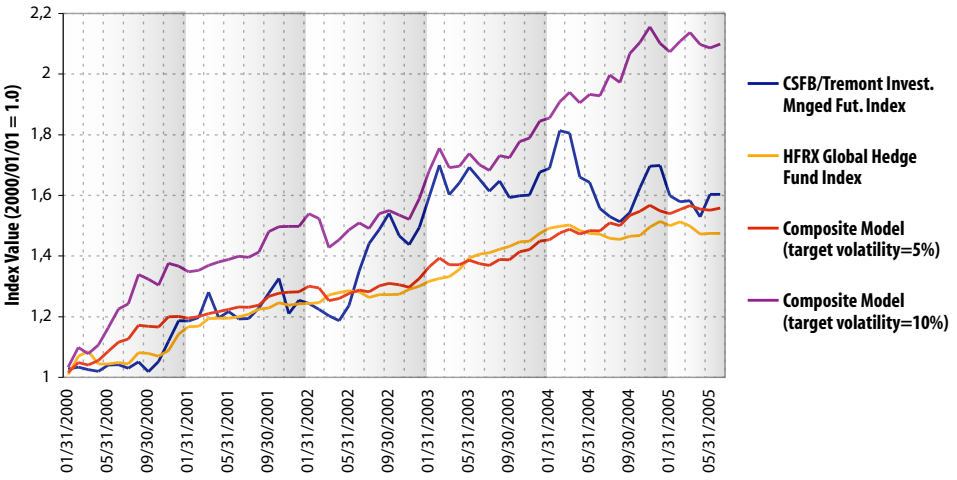
3. *Final Weight Normalization* (the weight attributed to each model by the mixture is given by $v_{t+1,k}$):

$$v_{t+1,k} = \frac{w^s_{t+1,k}}{\sum_j w^s_{t+1,j}}.$$

In these equations, η is a constant *learning rate* hyperparameter that controls the responsiveness of the weight adaptation. The α hyperparameter, along with the intermediate variable pool_t, define a "sharing mechanism" whereby more important models distribute a part of their weight to the less important models; this is important to prevent some models from having their weight decay to zero and enables the mixture to better track nonstationarities. Obviously, it remains to choose how to fix those parameters. In the spirit of "letting the data speak for itself", we run several of those mixtures in parallel, varying the η parameter between a "fast" and a "slow" mixture, and make the final determination by the following step.

An extensive analysis of the exponentiated gradient mixture, including bounds on the generalization error, is provided by Herbster and Warmuth (1998).

In terms of learning primitives, the mixture operates entirely in an online fashion, and requires no separate batch training step.



◀ **Figure 4.12.** *Cumulative return of the composite-learner-based strategy over the 2000–05 period, for two levels of annual target volatility (5% and 10%). For comparison purposes, the performance of two well-known hedge fund indexes are illustrated.*

▶ **D. Hyper-Mixtures :** As noted, in the exponentiated-gradient algorithm, hyperparameters control the convergence rate and the minimum value of a weight; these are not known *a priori*, so we run several such mixtures in parallel. The hyper-mixture step *selects* (rather than combines) the best-performing mixture over a one-year horizon. The selection is based on a financial performance criterion (the Sharpe (1966) ratio, which measures the average excess return over the risk-free rate unit of standard deviation of portfolio returns). Note that two separate selections are carried out, one for the technical and one for the fundamental model.

▶ **E. Exponentiated-Gradient Mixture :** Finally we must combine the technical and fundamental sides into a final single allocation. Again, since mixing weights are not known *a priori* another layer of mixtures (several of them since their hyperparameters are not known either) is used to form an adaptive combination.

▶ **F. Final Hyper-Mixture :** This final layer selects the best-performing mixture in the same spirit as step D, also on a one-year horizon, and considering a financial performance criterion.

In Figure 4.9, each vertex (except for the input variables) is considered a temporal learning algorithm in the sense of definition 7, even though the full ramifications of the definition are not necessary in all instances. (For instance the mixtures only perform online adaptation and do not require a training part.) This allows the algorithms of section 4.4 to be applied with no modification, resulting in a flexible and robust overall system. Figure 4.12 illustrates the simulated financial performance of the strategy embedded by the composite learner strategy.

4.7 Discussion

It is obvious that the functional composition of models is as old as statistical modeling itself. In the machine-learning literature, a number of composition mechanisms and "meta-algorithms" have been proposed, but much less attention has been paid to the engineering issues surrounding composition.

In parallel to this, machine-learning research overwhelmingly employs held-out tests to evaluate the out-of-sample performance of models and as a general framework allowing unbiased comparison between models. These approaches have been becoming more popular in empirical economics and finance in recent years as well. Excepting technical details such as those illustrated in Figure 4.4, out-of-sample testing provides a true unbiased estimator of future performance (under the strong assumption that the data-generating process is IID). For data in which ordering matters, *sequential validation* can be applied as a reasonable alternative.

We examined the issues surrounding composition in the presence of sequential validation. After introducing the challenges inherent in the sequential validation evaluation of learning algorithm performance, including the correct handling of revisions in economic time series, we introduced a definition of temporal learning that lends itself very well to a set of systematic algorithms to efficiently and correctly implement sequential validation in practical contexts. These algorithms take care of keeping each training set, and by extension the learners in a temporal learning network, up-to-date with respect to its sources; as such, they free the engineer to concentrate on the true difficulties of the problem s/he is facing, and not the incidental details of composing several models into a working system.

We illustrated the flexibility of the approach by introducing a portfolio allocation system made up of a number of learners, from the fixed decision rules, to batch learners, to online learners, and how they can be naturally combined to form a powerful decision-support system whose performance can be rigorously ascertained.

5 K-Best Paths Methods for Portfolio Management

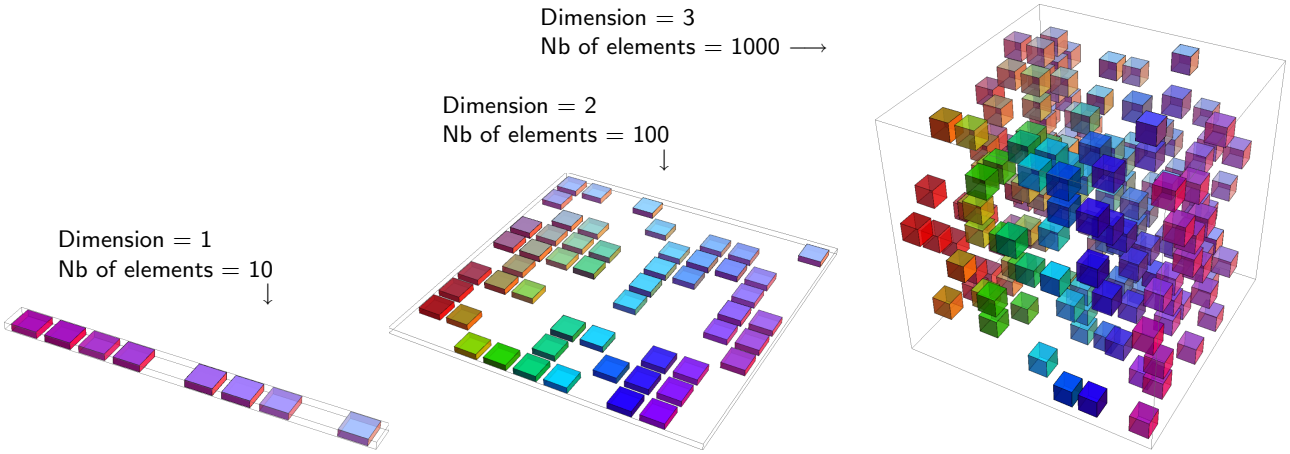
Life can only be understood backwards; but it must be lived forwards.

— *Søren Kierkegaard*

As explained at length in Chapter 2, it has long been known that the problem of optimal multiperiod portfolio choice can be expressed as a stochastic optimal control problem and solved by dynamic programming, following the classical solutions of Mossin, Samuelson and Merton (Mossin 1968; Samuelson 1969; Merton 1969). Such formulations assume that the investor is governed by additive utility functions, contingent on an immediate utility arising from intermediate consumption and on the distribution of terminal wealth. In practice, this may not be a realistic formulation: many risk-averse investors care as much about the portfolio *wealth trajectory* as they care about abstract higher moments of a conditional terminal wealth distribution. This explains the popularity of realized performance measures used by practitioners and professional fund managers, such as the Sharpe ratio (Sharpe 1966; Sharpe 1994), Information Ratio (Grinold and Kahn 2000), Sortino Ratio (Sortino and van der Meer 1991; Sortino and Price 1994) and Calmar Ratio. A common theme among these utility functions is that they depend on the entire sequence of realized returns (or statistics of the sequence). As such, they cannot conveniently be separated into a form amenable to solution by dynamic programming, unless one is prepared to suffer an inauspicious explosion in the size of the state space.

As noted in Chapter 3, dynamic programming is a general computational technique for solving sequential optimization problems that can be expressed in terms of an additive cost function (Bellman 1957; Bertsekas 2005). As is well-understood, it suffers from the so-called *curse of dimensionality*, wherein the computational cost of a solution grows exponentially with aspects of the problem dimension (size of the state, action and disturbance spaces); this phenomenon is depicted graphically in Fig. 5.1. Many approximation algorithms have been proposed in recent years for tackling large-scale problems, in particular by making use of simulation and function approximation methods. Still, most of these methods remain within the confines of traditional dynamic programming, which assumes that the function to be optimized can be separated as a sum of individual cost-per-time-step

Earlier versions of this chapter appeared as (Chapados and Bengio 2006) and (Chapados and Bengio 2007b)



▲ **Figure 5.1.** *The* ***curse of dimensionality****. As the dimensionality grows (here from one to three dimensions), more and more "boxes" are required to cover the space; this number grows exponentially with the dimensionality.*

terms and, for finite-horizon problems, a terminal cost. Unfortunately, for more complex utility functions, which may *depend on the trajectory of visited states*, classical dynamic programming and related approximations do not provide ready solutions.

One might argue that dynamic programming should be abandoned altogether, and one ought instead to revert to general nonlinear programming algorithms (Bertsekas 2000) to attempt optimizing under such utilities. This is the approach followed, in a sense, by Bengio's direct optimization of a financial training criterion (Bengio 1997), Moody's direct reinforcement algorithm (Moody and Saffell 2001b) and Chapados and Bengio's direct maximization of expected returns under a value-at-risk constraint (Chapados and Bengio 2001). However, these methods are found lacking in two respects: (i) they still rely on, either time-separable utilities (such as the quadratic utility), or on approximations of trajectory-dependent utilities that enable time-separability, (ii) they fundamentally rely on stochastic gradient descent optimization, and as such can be particularly sensitive to local minima. As we will see later, even a problem as simple as the optimization of a realized Sharpe ratio is made very difficult when transactions costs are introduced.

This chapter investigates a different avenue for portfolio optimization under general utility functions. It relies on formulating portfolio optimization on historical data as a deterministic shortest path problem, where we extract not only the single best path, but the K best paths, yielding, after some transformations, a training set to train a supervised learning algorithm to act as a controller. This controller can directly be used in a portfolio management task. This approach can be viewed as an instance of a "direct approach" to portfolio choice, some of which were reviewed in §2.3/p. 59.

The chapter is organized as follows: first we discuss the difficulties in using the Sharpe ratio as an optimization criterion rather than a comparison

criterion among fixed sequences of realized returns (§5.1/p. 123). Then, we introduce the overall approach (§5.2/p. 133), investigate in more detail the K best paths algorithm that we used (§5.3/p. 138), apply the algorithm to a portfolio setting and consider non-additive variants that yield *noisy* K best paths (§5.4/p. 142). Next, we introduce a learning architecture for making use of the K best paths targets through the framework of *ordinal regression* (§5.5/p. 161), outline experimental questions and methodological issues (§5.6/p. 169), present extensive experimental results demonstrating the value of the approach (§5.7/p. 173); and conclude (§5.8/p. 189). The appendices to this chapter provide some background on statistical techniques used in the analysis of results (§5.A/p. 192), give a complete specification of the input variables used in the experiments (§5.B/p. 197) and supply detailed results for all markets and models (§5.C/p. 212–§5.E/p. 233).

5.1 On Optimizing Sharpe Ratios

The Sharpe ratio (Sharpe 1966; Sharpe 1994) is frequently used by practitioners to assess the performance of investment alternatives.* Let R_t, $t = 1, \ldots, T$, be a sequence of realized portfolio returns over some horizon; the **empirical Sharpe ratio** is defined as

$$\mathrm{SR} = \frac{\hat{\mu} - R_f}{\sqrt{\hat{\sigma}^2}}, \tag{5.1}$$

with

$$\hat{\mu} = \frac{1}{T}\sum_{t=1}^{T} R_t, \qquad \hat{\sigma}^2 = \frac{1}{T-1}\sum_{t=1}^{T}(R_t - \hat{\mu})^2,$$

and R_f is the risk-free rate during the period (assumed to be constant, for simplicity). We call SR an *empirical* Sharpe ratio to emphasize that it is obtained from averaging a series of returns measured through time. It is common practice to annualize this ratio by the square-root-of-time rule: for instance, if the portfolio returns are obtained at a monthly frequency, the annualized ratio is obtained as $\sqrt{12}\,\mathrm{SR}$, since there are twelve months to a year.†

*And plays a role in determining the compensation of many a fund manager.

†This rule is derived under the assumption that the returns are IID. Consider a year's worth of (small) IID monthly random returns $\{r_t^{\text{monthly}}\}_{t=1}^{12}$. The annualized expected return is, for small returns, the sum of the individual returns,

$$\mathbb{E}[r^{\text{annual}}] = \mathbb{E}\Big[\sum_t r_t^{\text{monthly}}\Big] = \sum_t \mathbb{E}[r_t^{\text{monthly}}] = 12\mathbb{E}[r_t^{\text{monthly}}],$$

making use of the assumption of identical distribution. The annualized variance is

$$\mathrm{Var}[r^{\text{annual}}] = \mathrm{Var}\Big[\sum_t r_t^{\text{monthly}}\Big] = \sum_t \mathrm{Var}[r_t^{\text{monthly}}] = 12\,\mathrm{Var}[r_t^{\text{monthly}}],$$

The Sharpe ratio measures the portfolio return achieved in excess of the prevailing risk-free rate, per unit of assumed portfolio risk.* In a one-period mean-variance setting, the tangency portfolio, located on the efficient frontier, is that which maximizes the Sharpe ratio (see §A.2/p. 296). Because of this relationship with the well-understood quadratic utility problem, there has perhaps been too little interest paid to direct maximization of the Sharpe ratio.

Indeed, why should this even be a pertinent question, given its equivalence to a problem readily solved by the already-considerable apparatus of mean-variance optimization? To answer this question, one needs to distinguish between *expected* and *realized* performance measures. It is true that in the single-period setting, a rational (mean-variance) investor should aim, *ex ante*, at some combination of the risk-free asset and the tangency portfolio as this maximizes the expected Sharpe ratio for a given risk level, a consequence of classical separation theorems (§2.1.3/p. 12). Furthermore, this extends to the multiperiod case provided that returns are IID or unhedgeable (§2.2.3/p. 49).

However, consider the slightly different situation that is faced by the rational portfolio manager whose performance (and future career opportunities) is evaluated on the basis of the *realized* Sharpe ratio (5.1). Realized portfolio performance, the returns R_t, can only be known *ex post*; expectations—market views—that were held earlier are of no import when making this computation. As a result, the optimal behavior for the portfolio manager, *ex ante*, is to act as to yield a stream of realized returns that will, at the end of the horizon, maximize realized performance.

This latter problem is of a very different nature than straightforward mean-variance optimization. As we shall see below, it carries considerable numerical difficulties, particularly when higher-than-minuscule transaction cost levels are considered.

In fact, as shall become clear, the plain Sharpe ratio, by itself, may be an adequate criterion to *compare the realized performance* of several investments. However it is wholly ineffective as an *optimization objective*, when used, for instance, in training a learning algorithm. The reasons for this shortcoming, detailed next, include a propensity to diverge when optimizing weights over realized returns, and the absence of a preferred leverage scale, which makes several solutions equivalent.†

making use of both the identically-distributed and independence assumptions (independent variances add). Taking the square root of the variance to obtain a standard deviation, and using it as the divisor of the expectation to obtain a Sharpe ratio, we see that a $\sqrt{12}$ remains in the numerator, which serves as the proportionality constant between the monthly and yearly Sharpe ratios.

*As measured by the standard deviation of returns; see §2.1.5/p. 16 for other possibilities.

†*Financial leverage* is the ratio of the portfolio asset value (of, e.g., a portfolio or a firm) to the supporting equity; a leverage greater than 1 is achieved by borrowing the difference between assets and equity, i.e. by taking on debt. Leverage magnifies the return on equity

5.1.1 Regularization

Consider a multiperiod one-risky-asset problem where at each time-step t, the investor allocates a fraction w_t, $0 \leq w_t \leq 1$, of its capital to the risky asset, earning a (random) portfolio return $w_t R_t$; for simplicity, assume that the risk-free return earned on the remainder is zero.* Denote by $\mathbf{w} = (w_1, \ldots, w_T)'$ the sequence of weights and by $\mathbf{R} = (R_1, \ldots, R_T)'$ the random asset returns. For fixed weights $\mathbf{w}$, the regularized expected empirical Sharpe ratio is expressed as

$$\mathrm{SR}(\mathbf{w}; \epsilon) = \mathbb{E}_{\mathbf{R}}\Big[\frac{\hat{\mu}(\mathbf{w}, \mathbf{R})}{\sqrt{\hat{\sigma}^2(\mathbf{w}, \mathbf{R})} + \epsilon}\Big] \tag{5.2}$$

with

$$\hat{\mu}(\mathbf{w}, \mathbf{R}) = \frac{1}{T}\sum_{t=1}^{T} w_t R_t, \qquad \hat{\sigma}^2(\mathbf{w}, \mathbf{R}) = \frac{1}{T-1}\sum_{t=1}^{T}(w_t R_t - \hat{\mu}(\mathbf{w}, \mathbf{R}))^2$$

and ϵ a small non-negative quantity. Consider for now the case where no regularization is applied, $\epsilon = 0$. The expected empirical Sharpe ratio (5.2) is the natural criterion that a portfolio manager would want to maximize. It may serve as the basis for the *training objective* of a learning algorithm, a point that we revisit in §5.5.3/p. 165. In this context, the expectation in eq. (5.2) would be replaced by a sample mean over historical return realizations, and it is instructive to investigate the situations where the criterion may misbehave. Expanding the quantity $\hat{\sigma}^2(\mathbf{w}, \mathbf{R})$, which is part of the denominator in eq. (5.2), we have

$$\begin{aligned}\hat{\sigma}^2(\mathbf{w}, \mathbf{R}) &= \frac{1}{T-1}\sum_{t=1}^{T}\Big(w_t R_t - \frac{1}{T}\sum_{\tau=1}^{T} w_\tau R_\tau\Big)^2 \\ &= \frac{T}{T-1}\Big(\frac{1}{T}\sum_{t=1}^{T} w_t^2 R_t^2 - \Big(\frac{1}{T}\sum_{t=1}^{T} w_t R_t\Big)^2\Big).\end{aligned}$$

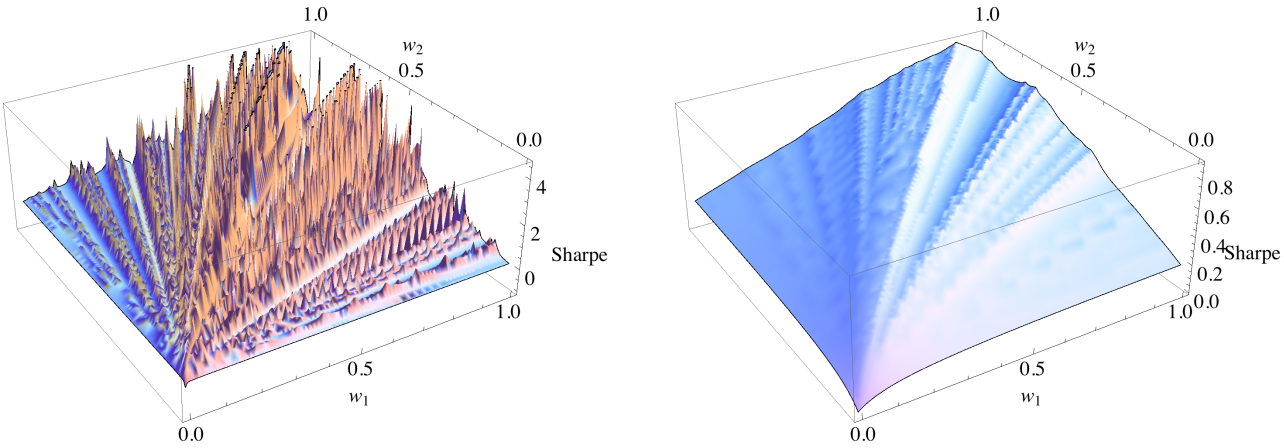
By inspection, this quantity is zero if

$$\sum_{t=1}^{T} w_t^2 R_t^2 = \frac{1}{T}\Big(\sum_{t=1}^{T} w_t R_t\Big)^2. \tag{5.3}$$

This condition causes problem for finite Monte Carlo approximations of eq. (5.2) since it needs only to hold for a *single realization* of the asset

at the cost of higher risk. The footnote on p. 296 illustrates why, in the absence of a sum-to-one constraint, portfolios at different leverage levels can exhibit the same Sharpe ratio, implying that the latter, when used as an optimization criterion, must be supplemented by an external specification of the desired portfolio leverage for the problem to be well-posed.

*Which is, as of December 28th 2008, a surprisingly accurate assumption!



▲ **Figure 5.2.** *Illustration of the need to regularize the Sharpe ratio: two-period empirical Sharpe ratio (eq. 5.2) as a function of asset weights* $\mathbf{w}_1$ *and* $\mathbf{w}_2$*, computed from 250 simulated trajectories of a VAR process.* ***Left:*** *With regularization coefficient* $\epsilon = 10^{-4}$ ***Right:*** *With regularization coefficient* $\epsilon = 10^{-2}$*. Due to singularities, the need for regularization is obvious.*

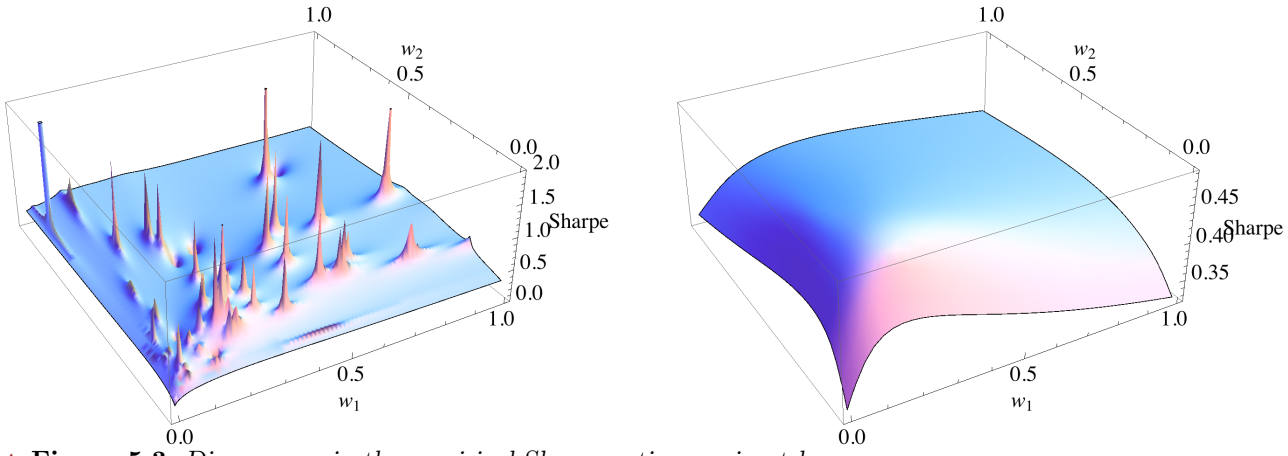
returns $\mathbf{R}$ for the entire summation to diverge. For instance, in the case $T = 2$, it suffices that

$$\frac{w_1}{w_2} = \frac{R_2}{R_1}$$

for this situation to arise; for longer horizons, the conditions are not so compactly written, but are seen from (5.3) to be satisfied at least for $w_t \equiv 0$. Although for any fixed $\mathbf{w}$, we would almost surely never draw an offending $\mathbf{R}$, the situation reverses when *optimizing* over $\mathbf{w}$, due to the systematic exploration over $\mathbf{w}$. This is the reason why it is essential to incorporate some regularization in the function $\mathrm{SR}(\cdot)$, when it is used as an optimization objective. One simple form is to incorporate a small constant in the denominator, as shown in eq. (5.2).

Figure 5.2 illustrates the need to regularize expected empirical Sharpe ratios. We consider here a small horizon ($T = 2$), where 250 independent draws from the VAR process (2.16) are taken with an initial dividend yield of 5%. The left part of the figure shows the result of eq. (5.2) as a function of the two portfolio weights, when a small regularization ($\epsilon = 10^{-4}$) is employed, whereas the right part shows the same function with a higher regularization ($\epsilon = 10^{-2}$). It is trusted that one should not need much convincing to conclude that the left plot constitutes a rather hopeless optimization objective.

At longer horizons, the problem remains but becomes harder to notice, and hence more pernicious. Figure 5.3 (left) shows the same objective function with $T = 3$, where the last weight is held fixed, $w_3 = 0.2$ and $\epsilon = 0$. Divergences in the objective remain but are more sporadic. At even longer horizons, it may be very difficult to visualize the problem; the right part of



▲ **Figure 5.3.** *Divergences in the empirical Sharpe ratio remain at longer horizons.* ***Left:*** *Three-period horizon, with the last asset weight w_3 held fixed at 0.2.* ***Right:*** *At even longer horizons, the problem may seem to disappear, but is merely hidden from view; here a five-period horizon with $w_3 = w_4 = w_5 = 0.2$ seems fine, but setting (e.g.) all five weights to zero produces a divergence.*

Fig. 5.3 shows the behavior of eq. (5.2) over a five-period horizon where the last three asset weights are held fixed at 0.2 (and $\epsilon = 0$). In this subspace, the objective (over w_1 and w_2 only) is very well behaved. This illustrates why, over long horizons, the unregularized objective can be treacherous: it can often be fine, and blow up for "no apparent reason" once in a while.

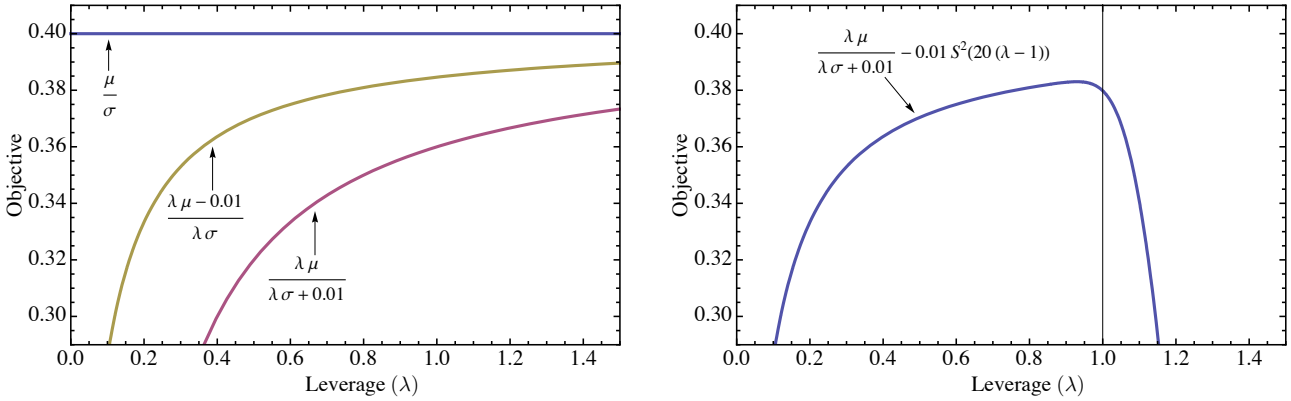
5.1.2 Leverage Scale

It is not sufficient to add regularization to obtain an adequate objective for maximization purposes. As shown on Fig. 5.4 (left), a plain Sharpe ratio without risk-free rate (μ/σ) lacks a preferred leverage scale (blue curve): multiplying portfolio positions by a constant λ leaves the ratio unchanged.* Subtracting a risk-free rate (yellow curve, denoted $\frac{\lambda\mu - 0.01}{\lambda\sigma}$) or adding regularization (red curve, denoted $\frac{\lambda\mu}{\lambda\sigma + 0.01}$) make the matter worse, since they both have *infinite* preferred leverage.

Of course, one can impose a hard leverage constraint by introducing a Lagrange multiplier in the objective function. However, such a penalty would typically symmetrically penalize departures from the target leverage. In many portfolio management contexts, one rather wishes for an asymmetric penalization: on the one hand, budget and risk management constraints prevent one from exceeding a leverage limit and must be quite forcefully enforced; on the other hand, a reduced leverage may occasionally be desirable if it allows one not to be exposed to adverse market movements.

For this reason, we propose a type of barrier penalty based on a squared

*This is equivalent to changing the risk level of the portfolio; it corresponds to moving along the Capital Market Line in Fig. 2.2 (p. 13).

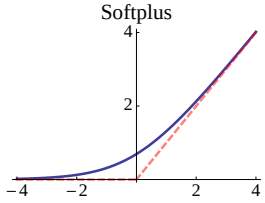


▲ **Figure 5.4.** ***Left:*** *A pure Sharpe ratio (μ/σ, taken, in this figure, to be $\mu = 0.1, \sigma = 0.25$) does not induce a preferred leverage scale on portfolio positions, yielding plateaux in the objective function landscape; subtracting a risk-free rate or adding a regularizing constant to the denominator (to prevent divisions by zero during optimization) do not resolve the issue, having* infinite *preferred leverage. The leverage parameter is given by λ.* ***Right:*** *A barrier penalty term given by the squared softplus function can induce a preferred leverage level; the location of the preferred leverage, $\lambda = 1$, is indicated by the vertical line.*

softplus function, where the complete objective takes the form

$$\mathrm{SR}(\mathbf{w}; \epsilon, \alpha, \beta, \bar{\lambda}) = \mathbb{E}_{\mathbf{R}}\Big[\frac{\hat{\mu}(\mathbf{w}, \mathbf{R})}{\sqrt{\hat{\sigma}^2(\mathbf{w}, \mathbf{R}) + \epsilon}} - \frac{\alpha}{T}\boldsymbol{\iota}'\mathrm{softplus}^2\big(\beta(\boldsymbol{\lambda} - \bar{\lambda})\big)\Big], \quad (5.4)$$

where $\boldsymbol{\lambda}$ is the vector of net leverage at all time-steps, $\boldsymbol{\lambda} = (|w_1|, \ldots, |w_T|)'$,* α and β are hyperparameters controlling the importance and shape of the barrier penalty, $\bar{\lambda}$ is the preferred leverage, and $\boldsymbol{\iota}$ is a vector of ones. The softplus function is a "smooth" version of the $\max(0, x)$ function,



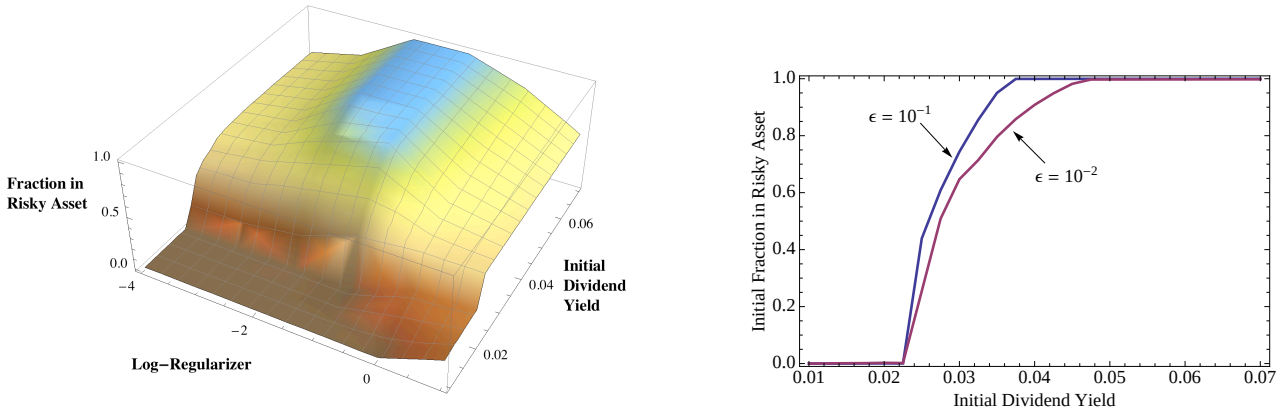
$$\mathrm{softplus}(x) = \log(1 + \exp(x)), \quad (5.5)$$

and we assume that it applies elementwise to vector arguments. Figure 5.4 (right) shows the impact of this penalty, where the parameters $\epsilon = 0.01, \alpha = 0.01, \beta = 20, \bar{\lambda} = 1$ are used.

5.1.3 Optimal Policies

To gain insight into the optimal behavior under the regularized Sharpe ratio objective, we carry on with the numerical simulation presented in §2.2.1/p. 42 for the power utility. Using the same VAR process (2.16) for generating asset returns, we draw 2500 independent realizations of a 10-period process, and optimize eq. (5.2)—where the expectation is replaced by a sample mean over the sampled trajectories—for several regularization

*In a multi-asset context, each element would be the sum of net exposures, i.e. $\boldsymbol{\lambda}_t = \sum_i |w_{t,i}|$.



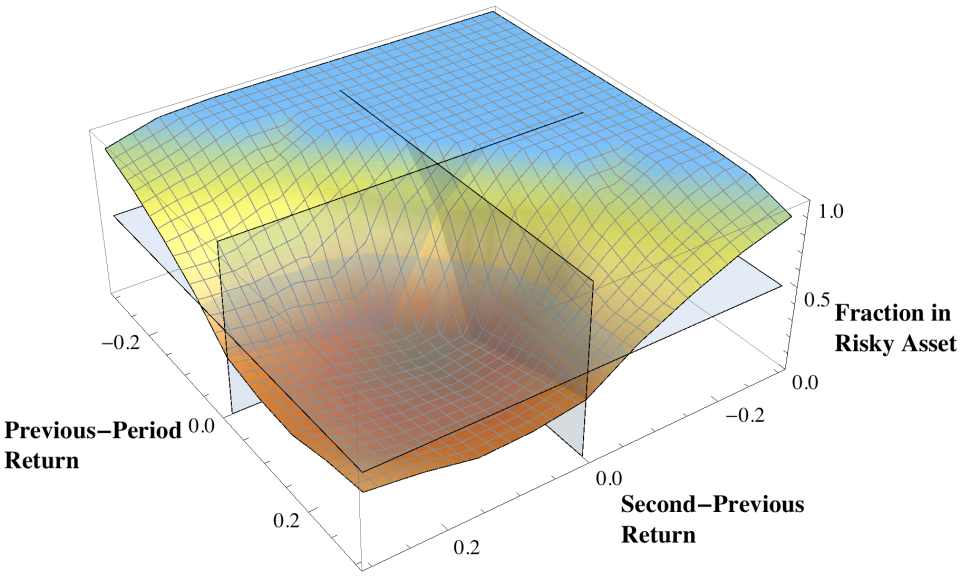
▲ **Figure 5.5.** ***Left:*** *Fraction of wealth initially invested in the risky asset for the two-asset problem under the regularized Sharpe ratio objective, as a function of the initial dividend yield and regularization coefficient (10-period horizon)* ***Right:*** *Two slices of the left plot, at different regularization coefficients.*

coefficients ϵ and initial dividend yields (which affect the generating process). The weights at each time-step were constrained to lie within the interval $[0, 1]$ and hence the more complex objective (5.4) was not employed.

Figure 5.5 shows the optimal initial policy (i.e. what should be the allocation at $t = 1$) obtained when carrying out stochastic gradient optimization with multiple restarts (to avoid getting trapped in local optima). Comparing the policy to that obtained under the power utility (Fig. 2.4, p. 43), we note that the allocation to the risky asset starts at a lower initial yield, and behaves as though the risk aversion *increases* as the initial yield increases. In other words, we reach the full allocation to the risky asset more slowly than under the power utility that would behave similarly at low initial yields.

However, the most significant difference between the Sharpe ratio objective and the power utility lies in the behavior at later timesteps. Recall that the value function under the power utility may be decomposed into a product of two factors, the first one depending on wealth only and the second on the state variables and remaining horizon. In particular, the optimal action at each time depends only on the latter, but not on current wealth.

In contrast, the optimal action under the Sharpe ratio objective depends on previous return realizations. This is illustrated in Fig. 5.6, where the optimal action at the third time-step (for a 10-period problem, using the same conditions as previously) is plotted as a function of the return realizations in the previous two time-steps. We observe that the allocation to the risky asset is greatly reduced if the previous two portfolio returns were positive; conversely, the maximum risk is taken if the previous two returns were negative. This would correspond to a "buy-low / sell high" rule: if the investor did well in the earlier timesteps, she would reduce her exposure to the risky asset and prefer more cash; in contrast, early negative returns would lead



▶ **Figure 5.6.** *Optimal policy (allocation in the risky asset) under the Sharpe ratio utility for the third time-step of a ten time-step problem, given the previous two time-step realizations; the policy depends on the realized returns: two good returns (front corner) brings a significant increase in the effective risk aversion and decrease exposure to the risky asset. The translucent cutting planes serve as grid lines.*

to a later increased exposure, the investor then betting on a reversion of returns to the mean.

This result carries implications for portfolio managers. Cvitanić, Lazrak, and Wang (2008) studies optimal Sharpe policies and recover the above-noted effects that the risk taken by the optimal policy is negatively correlated with past performance, an effect first observed by Brown, Harlow, and Starks (1996).*

5.1.4 Difficulty of Optimizing Realized Sharpe Ratios by Gradient Descent

The previous simulations were all carried out without accounting for transaction costs, a reality that is regrettably faced by all practical implementations. Optimization under transaction costs is a substantially more difficult problem, and is the main reason for the K-best paths methods that are introduced later in this chapter.

For now, we simply illustrate the difficulty of optimizing Sharpe ratios with standard gradient-based algorithms when transaction costs are involved. To this end, we carried out a simulation study wherein we optimize a sequence of decisions to maximize a Sharpe ratio, varying the level of transaction costs. More specifically, across several markets and time periods, and a wide range of transaction cost levels, we randomly initialize 100 sequences of market positions, which we use as starting points for a conju-

*Although the latter authors propose an alternative explanation in terms of a "tournament" between funds with similar investment objectives, such that there is a strong incentive for mid-period likely losers to take on additional risk to increase their end-period relative standing among their peers.

gate gradient optimizer (Bertsekas 2000) to maximize the realized Sharpe ratio over the time period. We then examine the distribution of resulting 100 Sharpe ratios after optimization, and investigate how variable this distribution becomes as we increase transaction costs.

Since, for a given combination of time-period/market/transaction-costs, the only factor of variability in the simulation is the random initialization of market positions taken at each time step, it follows that any variance observed after optimization depends on the random starting point. Because the optimized function remains the same, this variance can only be a consequence of local maxima in the optimization objective.

The following parameters are varied:

- **Market**: major stock market indices, the CAC 40 (France), DAX 30 (Germany), TOPIX (Japan), TSX 60 (Canada), and Russell 1000 and S&P 500 (both United States);
- **Time periods**: yearly time horizons, from 2003 to 2006.
- **Transaction costs**: from 0 to 250 BP* in increments of 25 BP.

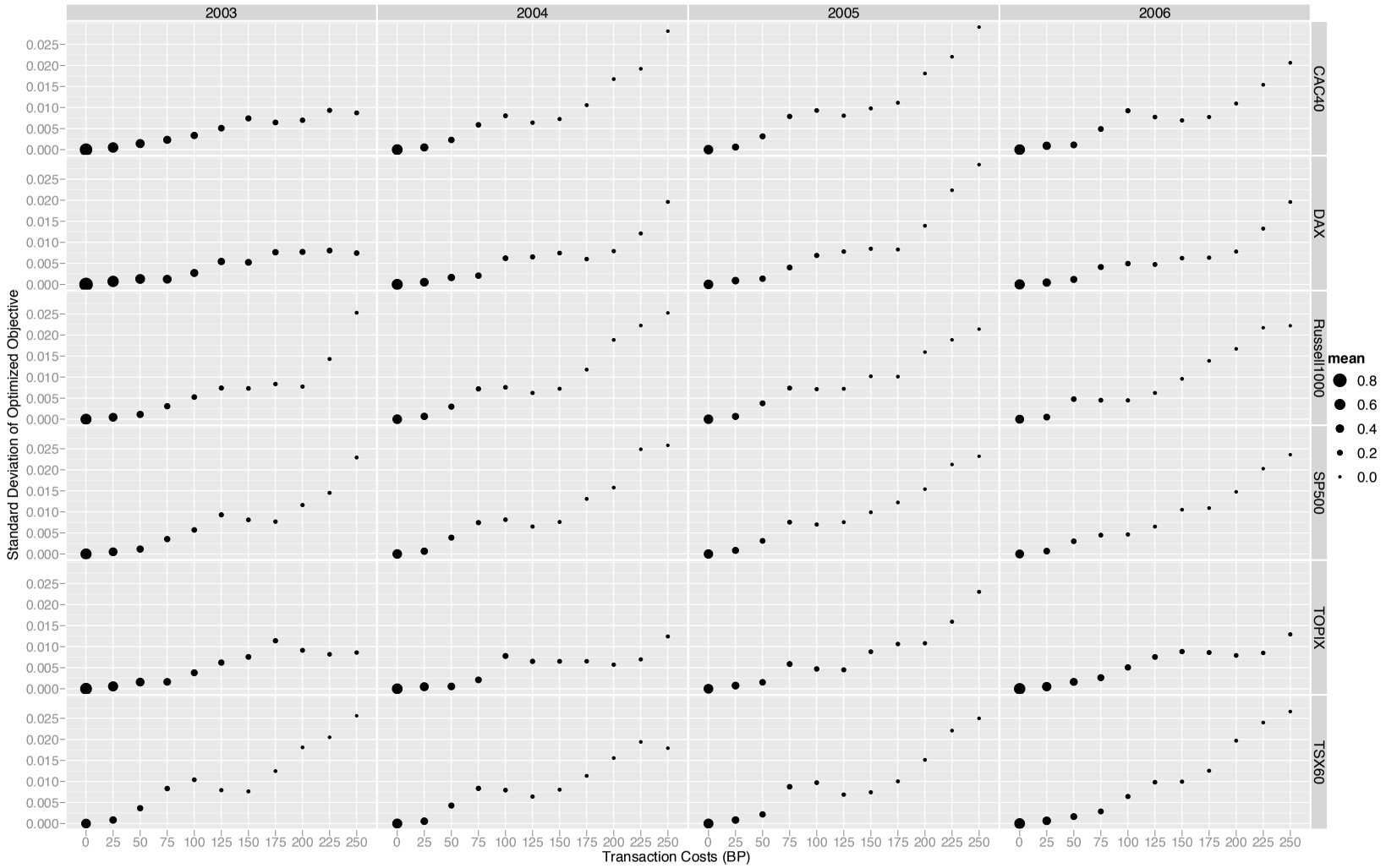
*Basis Point (one hundredth of one percent).

For each combination of those parameters, 100 initial starting points (each representing decisions to be made for one year of trading, representing approximately 250 decision variables) are drawn randomly from an isotropic normal distribution with a mean of zero and a standard deviation of 0.5.

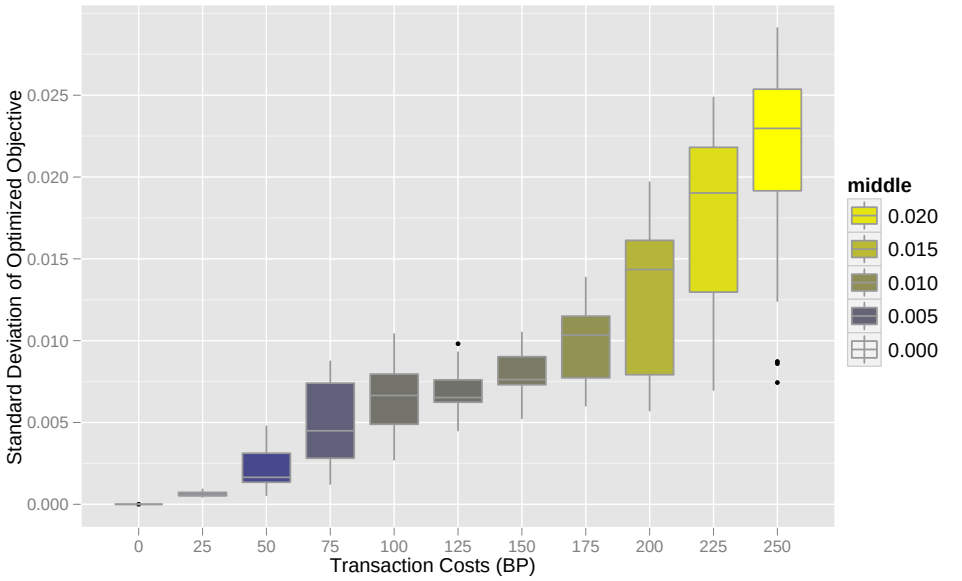
The standard deviation of the resulting Sharpe ratios at optimum is shown in Fig. 5.7, displayed as a function of the transaction costs, year and market. The mean Sharpe ratio is also represented by the size of the dot, and is seen to uniformly decrease with transaction costs. The plot clearly shows that, very reliably, the empirical standard deviation of Sharpe ratios after optimization increases with transaction costs, indicating that the issue of local minima becomes more severe as costs increase. This can be understood by noting that, as costs increase, the optimization problem becomes increasingly combinatorial in nature, with decision variables bound over progressively longer horizons.

A more compact understanding can be reached from Fig. 5.8, which summarizes as boxplots the *distribution of Sharpe ratio standard deviations* at optimum across the markets and periods covered in Fig. 5.7, as a function of transaction costs. We clearly note, as previously, the trend of increasing standard deviations with increasing costs; moreover, we now clearly see that the *spread of the distribution* increases as well, suggesting that the problem becomes "unpredictably" more difficult.†

†Although a detailed analysis would take us far outside the scope of this work, this behavior suggests analogies to phase transition phenomena in combinatorial optimization, first noticed in random instances of the satisfiability (SAT) problem (Mitchell, Selman, and Levesque 1992; Kirkpatrick and Selman 1994), but now known to be present in a large number of problems, including the vertex cover and travelling salesman problem (Hartmann and Weigt 2005). Very deep similarities exist between these phenomena and



▲ **Figure 5.7.** *Variance in Sharpe ratios optimized by conjugate gradient descent, as transaction costs are varied from 0 to 250 basis points, across years (columns) and markets (rows). The mean Sharpe ratio is represented by the size of each dot. The variance increases almost uniformly with costs, suggesting the presence of local minima.*



◀ **Figure 5.8.** *Distribution of the* ***standard deviation*** *of optimized Sharpe ratios, pooling together all results from Fig. 5.7, as a function of transaction costs. With increasing costs, both the mean standard deviation increases, but its own variance increases as well. Boxplots are filled with a color progression that depends on the location of the median.*

From these results, one is compelled to conclude that, as soon as even small levels of transaction costs are involved, using a gradient-based method to optimize a sequence of decisions to maximize a Sharpe ratio is a terrible idea. Even worse, the cases covered here are "easy" one-asset problems. For portfolios involving multiple assets, the combinatorics of local minima becomes exponentially more difficult.

5.2 Problem Formulation

We consider a discrete-time system in an observable state $\mathbf{x}_t \in \mathbb{R}^N$ at time t, which must select an action $\mathbf{u}_t \in \mathbb{R}^M$ at every time step. The system evolves according to a state-transition equation $\mathbf{x}_{t+1} = f_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t)$, where $\mathbf{w}_t$ is a random disturbance. Note that state transition is considered to be deterministic, assuming knowledge of the random disturbance $\mathbf{w}_t$. At each time-step, the system experiences a random reward $g_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t)$. Our objective is to maximize an expected utility U of the sequence of received

the behavior of spin glasses and the Ising model in solid-state physics (Mézard, Parisi, and Virasoro 1987). Although applications of the Ising model have been made to economics, including modeling the dynamics of agents' opinion in the presence of learning (Zhou and Sornette 2007) and to understanding market bubbles and crashes (Kaizoji, Bornholdt, and Fujiwara 2002; Sornette 2003), to the author's knowledge there has been no work on its possible connections with multiperiod financial portfolio optimization in the presence of transaction costs.

rewards over a finite horizon $t = 0, \ldots, T$,

$$J_0^*(\mathbf{x}_0) = \max_{\mathbf{u}_0,\ldots,\mathbf{u}_{T-1}} \mathop{\mathbb{E}}_{\mathbf{w}_1,\ldots,\mathbf{w}_{T-1}} \left[U(g_0, g_1, \ldots, g_T) | \, \mathbf{x}_0\right] . \tag{5.6}$$

Obviously, if $U(g_0, g_1, \ldots, g_T)$ can be written as $\sum_t g_t$, the finite-horizon problem is solved by writing the *value function* $J_t(\mathbf{x}_t)$ in terms of Bellman's equation,

$$J_T^*(\mathbf{x}_T) = g_T(\mathbf{x}_T) \tag{5.7}$$

$$J_t^*(\mathbf{x}_t) = \max_{\mathbf{u}_t} \mathop{\mathbb{E}}_{\mathbf{w}_t} \left[g_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t) + J_{t+1}^*(f_t(\mathbf{x}_t, \mathbf{u}_t, \mathbf{w}_t))\right] . \tag{5.8}$$

From the value function, the optimal action $\mathbf{u}_t^*$ at time t is obtained as that reaching the maximum in the equation above.

5.2.1 Solving for a General Utility

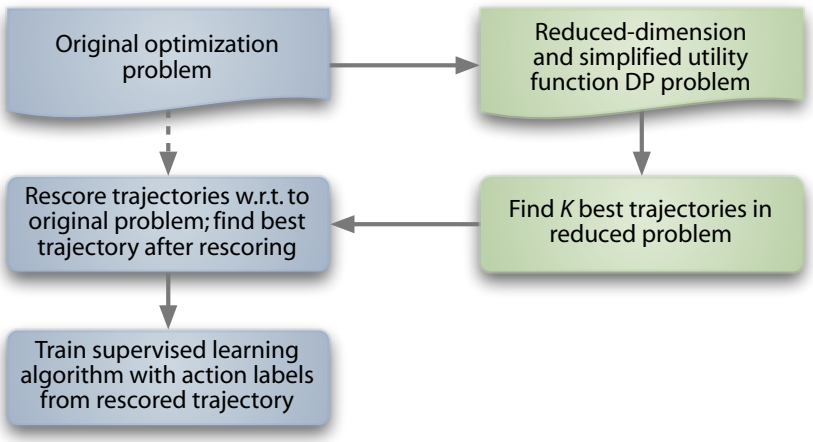
Our objective is to devise an effective algorithm to obtain optimal actions in the case of a general utility function. Although no recursion such as Bellman's can readily be written in the general case, a key insight lies in the simple observation that, given a **realized trajectory** of rewards, most utility functions (at least those of interest, for instance, in finance) can be computed quickly, in time $O(T)$. Hence, if we are given K such trajectories, we can find the best one under a *general utility function* U in time $O(K(T + \log K))$.

A second observation is that given this sequence of actions, we have obtained what amounts to a set of $\langle \text{state}_t, \text{action}_t \rangle$ pairs at each time-step within the trajectory. We can make use of these as a *training set for a supervised learning algorithm*. In other words, we can bypass completely the step of estimating a value function under the desired utility function, and instead directly train a controller (also called an *actor* in reinforcement learning (Sutton and Barto 1998)) to make decisions.

The two preceding observations can be combined into the following algorithm for approximately solving eq. (5.6):

1. **Generate** a large number of candidate trajectories;
2. **Rescore** (sort) the trajectories under the desired utility function U;
3. Use the best rescored trajectory to **construct a dataset** of all $\langle$state, action$\rangle$ pairs within the trajectory; carry out steps 1–3 until the dataset is large enough.
4. Using the dataset from steps 1–3, **train a supervised learning algorithm** to output the action label given the input state.

In this algorithm, the generation of trajectories (step 1) can be stochastic; however, the subsequent steps rely on the generated trajectories and as such, are deterministic.



◀ **Figure 5.9.** *Summary of the proposed algorithm for finding good trajectories under a non-additive utility function.*

As is common practice in reinforcement learning (Bertsekas and Tsitsiklis 1996), the algorithm estimates the expectation in eq. (5.6) with a sample average over a large number of trajectories. Furthermore, as we shall see below, for portfolio management applications, we can dispense with a generative model of trajectories by using historical data.

5.2.2 Generating Good Trajectories

It remains the question of *generating good trajectories* in the first place. This is where a K-best paths algorithm is involved: under an "easier" (i.e. additive) utility function acting as a proxy for our target utility, and over a large historical time period (which will become the training set), we use the K-best paths algorithm to generate the candidate trajectories of step (1) above. Obviously, both the "easier" and desired utility functions, henceforth respectively called the *source* and *target* utilities, must be correlated, so that searching for good solutions under one function has a high likelihood of yielding good solutions under the other. We discuss this point more fully below. Figure 5.9 illustrates schematically the complete algorithm.

5.2.3 Known Uses

This algorithm is certainly not the first one to make use of a K-best paths algorithm: they have been used extensively in speech recognition and natural language processing (e.g. Rabiner and Juang 1993). However, in these contexts, the rescored action labels found by the K-best paths are either discarded (speech) or not used beyond proposing alternative hypotheses (NLP). In particular, no use is made of the rescored trajectories for training a controller.

5.2.4 Relationship to Approximate Dynamic Programming Methods

Referring to the approximate dynamic programming methods of §3.2/p. 77, the approach proposed here is closest to the direct policy learning techniques (§3.2.4/p. 86). Although most of these methods optimize the policy by gradient ascent in policy parameter space with respect to a value function-related performance criterion, the K-best paths technique explicitly obtains the target actions along each sample trajectory (which can be obtained from historical data or Monte Carlo simulation), and uses those as part of the training set for a supervised learning algorithm.

Two compelling relationships exist between the currently proposed approach and the literature. First, several papers have explored the idea of using *reductions* to convert a reinforcement learning problem into a supervised classification problem. Lagoudakis and Parr (2003) introduce an approximate policy iteration method, in which the inner loop generates sample trajectories from the current policy and a rollout scheme* to construct a training set for a classifier (the authors experimented with both an SVM and a neural network); the resulting trained classifier is then used as the "improved policy" in the next iteration of the algorithm. More recently, Langford and Zadrozny (2005) established a formal connection between the prediction ability of a classifier and the performance of a general reinforcement learning algorithm (which is defined as to encompass both MDP†s and POMDPs as special cases). However, Langford *et al.* do not show how to construct the classifier training sets, but suggest using the trajectory tree method of Kearns, Mansour, and Ng (2000) if a generative model is available. The algorithm by Bagnell, Kakade, Ng, and Schneider (2004) is also similar in spirit to Langford *et al.*'s, although the former prove slightly weaker theoretical results (but produce experimental confirmation of their algorithm's validity on both MDPs and POMDPs).

†*Markov Decision Process.*

A second strong link can be drawn with the PEGASUS algorithm of Ng and Jordan (2000). PEGASUS uses Monte Carlo trajectories with common random numbers to ensure that all policies are evaluated with respect to the same realizations. This can be considered analogous to our approach where a number of sample trajectories are drawn once (and held fixed), then the target actions are found for each (from rescoring under the target utility), and finally a supervised learning algorithm is trained to best approximate this training set.

All of the proposed approaches, to our knowledge, have so far focused on staying within an additive utility function framework and assume the presence of a generative model to construct trajectory histories. That non-

*"Rollout" simply means that a number of finite-length trajectories are generated from a fixed starting state–action pair under a given policy to approximate the Q-values corresponding to the state–action pair.

additive utility functions could be of interest does not strike as a major concern of the current literature.

5.2.5 Limitations

An obvious and incompletely answered issue with the proposed K-best paths approach, as formulated here, is that it implicitly assumes that the historical trajectories provide sufficient examples to explore the state space well. In nonstationary situations—prevalent in finance—, this may result in a serious impediment to implementing the approach. It is also hard to relate, other than experimentally, the properties that a source utility should possess in order to provide a good set of trajectories after rescoring under the target utility.

This lack of guarantees prevents, at this juncture, proving strong theoretical performance bounds, and may also limit the practical performance of the algorithm. Historical trajectories may need to be supplemented by simulation approaches to generate a good coverage* and experimentation is required to ensure a good match between specific source and target utilities. We investigate some of these questions in later sections.

Nevertheless, due to its close links to algorithms for which theoretical properties have been established, in particular the reduction of Langford and Zadrozny (2005), the proposed K-best paths approach is justified in principle; determination of its applicability to particular domains ultimately becomes an experimental question.

5.2.6 Relationship with POMDPs

It should be stressed that the training set constructed after rescoring can contain far more information, in the input variables, than that used during the source-utility dynamic programming search and subsequent rescoring under the target utility. As we shall see in the experimental section, a large number of explanatory variables, each summarizing additional problem dimensions or aspects of the past, can be provided as input to help improve prediction performance.

The key to being capable of using these variables is that the actions taken by the controller must have no effect on them (for otherwise they should be part of the state space in the initial source-utility DP search); in other words, this approach is particularly well-suited when the state variables can be thought of as being made up of a large number of uncontrolled dimensions, such as the exogenous information that is obtained on financial markets.

Moreover, the controller can be an arbitrary sequential learning algorithm, such as a recurrent neural network. Provided that one is capable of training such networks, which is known to be difficult for gradient-based learning (Bengio, Simard, and Frasconi 1994), simple forms of POMDPs could

*Although we strictly rely on historical data in the results presented herein.

be tackled by this approach. In such models, the state is not completely observed and the controller must progressively build a representation of the "true state" through repeated interaction with the environment; in this context, a recurrent neural network (whose inputs consist of just the observable portion of the state) could make use of the storage capabilities of the recurrent units to assemble, given a sequence of inputs, a fuller picture of the true state.

5.3 Enumerating the K Best Paths

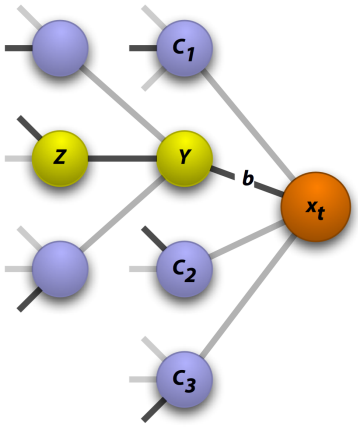
The problem of computing the K shortest paths between two vertices s and t in a directed graph $G = (V, E)$, with n vertices and m edges, has received much attention. While it is not our purpose to exhaustively survey every proposed solution, good references to the early literature are provided by Dreyfus (1969), and more recent ones by Eppstein (1998).* The fastest known algorithm in the worst case, asymptotically, is due to Eppstein (1998), who uses an implicit representation of deviations from the shortest path to compute the K shortest paths in order of increasing length in time $O(m+n\log n+K)$; he also solves the related problem of computing K paths from s to t that are shorter than a length limit L in time $O(m+n+K)$ once a shortest path tree in the graph is computed. Unfortunately, the implementation of Eppstein's algorithm is made complex by the need to maintain intricate data structures; furthermore, some computational results suggest that even for rather large graphs, other approaches can be faster in practice than Eppstein's (Jiménez and Marzal 1999).†

We rely on a time- and memory-efficient implementation of the Recursive Enumeration Algorithm (REA) of Jiménez and Marzal (1999), which has a worst-case time bound of $O(m + Kn\log(m/n))$. This algorithm is made effective by implicitly constructing a path from its *differences* with a previous path. It builds upon a generalization of Bellman's recursion of eq.(5.8) to a statement of optimality of higher-order paths in terms of lower-order ones. Before introducing a precise statement of the algorithm, an intuition into its operation can be obtained from Figure 5.10:

- Suppose that the best path to a vertex x_t ends with $(\ldots, Z, Y, x_t)$.
- According to the REA recursion, the **second best path** up to x_t is given by the best of:

*David Eppstein also maintains an online bibliography on "K-Best" problems at `http://www.ics.uci.edu/~eppstein/bibs/kpath.bib`.

†More recently, Jiménez and Marzal (2003) introduced a lazy version of Eppstein's algorithm that preserves the worst-case asymptotic time complexity but give it better performance in practice.



◀ **Figure 5.10.** *Intuition behind the recursive relationship underlying the REA K-best-paths algorithm; see text for details.*

1. Either the **first best path** up to the immediate predecessors of x_t, namely the *candidate vertices* $\{C_1, C_2, C_3\}$, followed by a transition to x_t.
2. Or the **second best path** up to Y, followed by the transition b to x_t. The second best path to Y is found by applying the algorithm recursively.

We now examine the generalization of Bellman's equations that specify the best paths after the first one, and state more formally the REA algorithm.

5.3.1 Generalized Bellman's Equations

We shall let each edge $(u, v) \in E$ be weighted by a function $\ell(u, v)$ giving the *length* of the edge. For simplicity, we shall assume that graph G does not contain negative edge lengths or cycles (which is appropriate given the graphs arising from the dynamic programming problems that we shall consider). The set of *predecessors* of vertex v is given by $\Gamma^{-1}(v) \triangleq \{u : (u, v) \in E\}$. For any $u, v \in V$, a path from u to v is denoted by

$$\pi = \pi_1 \cdot \pi_2 \cdot \cdots \cdot \pi_{|\pi|},$$

where $\pi_1 = u$, $\pi_{|\pi|} = v$, and all consecutive vertices belong to the set of edges, $(\pi_i, \pi_{i+1}) \in E$. Let the length of a path π be denoted by

$$L(\pi) \triangleq \sum_{1 \le i < |\pi|} \ell(\pi_i, \pi_{i+1}),$$

and for convenience $L(\pi) = 0$ if $|\pi| = 1$. We shall consider a fixed starting vertex s and terminal vertex t. Let $\pi^k(v)$ denote the k-th shortest path from the starting vertex s to an arbitrary v, and let $L^k(v) \triangleq L\big(\pi^k(v)\big)$ denote its length. Note that finding $\pi^1(v)$ corresponds to solving the well-known single-source shortest-path problem.

For each vertex $v \in V$, we maintain a set of *candidate paths* $C^k(v)$ providing possible solutions to $\pi^k(v)$. These sets, along with the associated $\pi^k(v)$ and $L^k(v)$ are defined recursively as follows.* The shortest-path $k = 1$ is a special case and an implementation of the algorithm (see below) solves it separately. See, e.g., Bertsekas (2005) for a variety of algorithms well-suited to this task.

Theorem 9 (Jiménez and Marzal) *For all $v \in V$,*

$$L^k(v) = \begin{cases} 0, & \text{if } k = 1 \text{ and } v = s; \\ \min_{\pi \in C^k(v)} L(\pi), & \text{otherwise}; \end{cases} \tag{5.9}$$

$$\pi^k(v) = \begin{cases} s, & \text{if } k = 1 \text{ and } v = s; \\ \arg\min_{\pi \in C^k(v)} L(\pi), & \text{otherwise}; \end{cases} \tag{5.10}$$

where if $k = 1$ and $v \neq s$, or $k = 2$ and $v = s$, then the candidates to v are obtained as

$$C^k(v) = \{\pi^1(u) \cdot v : u \in \Gamma^{-1}(v)\}; \tag{5.11}$$

otherwise, let u and k' be respectively the node and index that complete the previous best path to v, such that $\pi^{k-1}(v) = \pi^{k'}(u) \cdot v$, then the candidates to v are obtained recursively as

$$C^k(v) = \left(C^{k-1}(v) - \{\pi^{k'}(u) \cdot v\}\right) \cup \{\pi^{k'+1}(u) \cdot v\}, \tag{5.12}$$

where we assume that if the set of candidates $C^k(u)$ becomes empty, then $\pi^k(u)$ does not exist and all sets arising from concatenations of the form $\{\pi^k(u) \cdot v\}$ are empty.

Before stating the proof, it is useful to examine more closely the general recursive cases of eq. (5.9)–(5.10) and (5.12). The last one provides the key to the recursion: it *updates* the set of candidates for the k-th best path ending at vertex v as follows:

- It removes the just-extracted $k-1$-th best path from the possible candidates for the k-th best path, and
- it substitutes the next-worst path ending at the *same predecessor* vertex as the $k-1$-th best path.

The first two equations simply state that the k-th best path ending at v and its length are obtained by taking the best path within the current set of candidates for v.

Proof For $k = 1$, the recursion base case, the specification of the shortest path to any vertex v, $\pi^1(v)$, is equivalent to the well-known Bellman recursion. Now consider the case $k > 1$. Let $\mathcal{P}^k(u) = \{\pi^j(u) : j \leq k\}$ denote the

*See Bellman and Kalaba (1960) and Dreyfus (1969) for closely related formulations.

set of the k shortest paths ending with u. For $v \neq s$, the k-th shortest path must have a predecessor vertex u such that $u \in \Gamma^{-1}(v)$ and the resulting path not already in $\mathcal{P}^{k-1}(v)$; denote this path by $\pi^k(v) = \pi^{k'}(u) \cdot v$, for some k'. It is easy to see that only the smallest k' such that $\pi^{k'}(u) \cdot v \notin \mathcal{P}^{k-1}(v)$ needs to be considered since nonnegative edge lengths imply that

$$\tilde{k} > k' \Longrightarrow L\big(\pi^{\tilde{k}}(u)\big) + \ell(u,v) \geq L\big(\pi^{k'}(u)\big) + \ell(u,v).$$

Hence, in order to obtain the k-th best path ending at v, it is necessary to consider only $|\Gamma^{-1}(v)|$ predecessor paths, and for each one, keep track of the smallest k' that did not lead to a path in $\mathcal{P}^{k-1}$.* ■

5.3.2 Recursive Enumeration Algorithm and Data Structures

Listings 5.1 and 5.2 state the REA procedure in more detail. We assume that an (unspecified) procedure ComputeShortestPaths(G, s) is available to compute the first best paths $\pi^1(v)$ for all $v \in G$.† The first listing finds the K best paths ending at vertex t by enumerating them in sequence. We denote the condition that the path $\pi^k(v)$ does not exist by the symbol $\perp$.

Listing 5.1: Recursive Enumeration Algorithm

```
1 def RecursivelyEnumerateKShortestPaths(G, s, t, K):
2     ComputeShortestPaths(G, s)
3     for k = 2, ..., K:
4         if π^{k-1}(t) = ⊥ :
5             break
6         π^k(t) ⟵ NextPath(t, k)
```

The heart of the recursive enumeration procedure is NextPath(v, k), which returns the k-th best path ending at v; it calls itself recursively to update the set of candidate paths for vertex v and follows a quite direct implementation of eq. (5.12).

Listing 5.2: Computing the Next Path

```
1 def NextPath(v, k):
2     if k=2:
3         # Initialize the candidate set
4         C[v] ⟵ {π^1(u) · v : u ∈ Γ^{-1}(v), π^1(v) ≠ π^1(u) · v}
5     if v ≠ s or k ≠ 2:
6         # Update the candidate set
7         let u, k' be such that π^{k-1}(v) = π^{k'}(u) · v
8         if π^{k'+1}(u) does not exist:
9             π^{k'+1}(u) ⟵ NextPath(u, k' + 1)
```

*The proof is from Jiménez and Marzal, with some simplifications by the author.

†In our implementation, we make use of a beam search to compute approximate first best paths; see §5.4.5/p. 150.

```
        if π^{k'+1}(u) ≠ ⊥:
            C[v] ⟵ C[v] ∪ π^{k'+1}(u) · v
    # Extract a new path if the candidate set is not empty
    if C[v] ≠ ∅:
        π* ⟵ arg min_{π∈C[v]} L(π)
        C[v] ⟵ C[v] − {π*}
        return π*
    else:
        return ⊥
```

A prototypical data structure fragment for the algorithm is shown in Fig. 5.11. Each graph vertex (circles) contains a pointer to its first best path; successively worse paths ending at that vertex are kept as a linked list of path structures (green rounded rectangles). Each path structure contains a pointer to its associated ending vertex. Furthermore, each contains a back-pointer to the subpath ending at the previous vertex. The figure represents the three best paths ending at t, respectively: (1) $\pi^1(w) \cdot t$, (2) $\pi^2(w) \cdot t$, (3) $\pi^1(v) \cdot t$. Furthermore, the set of candidates for the fourth best path ending at t is shown; for efficiency, this set can be implemented with a priority queue of a more sophisticated data structure than the list shown here. Jiménez and Marzal (1999) provide a complete time and space complexity analysis of the algorithm.

5.4 Application to Portfolio Optimization

The portfolio optimization setting that we consider is a multi-period, multi-asset problem with transaction costs. We assume that the assets (e.g. stocks, futures) are sufficiently liquid that market impacts can be neglected. We invest in a universe of M assets, and the state $\mathbf{x}_t$ at time t is given by

$$\mathbf{x}_t = \begin{bmatrix} \mathbf{n}_t \\ \mathbf{p}_t \end{bmatrix} \tag{5.13}$$

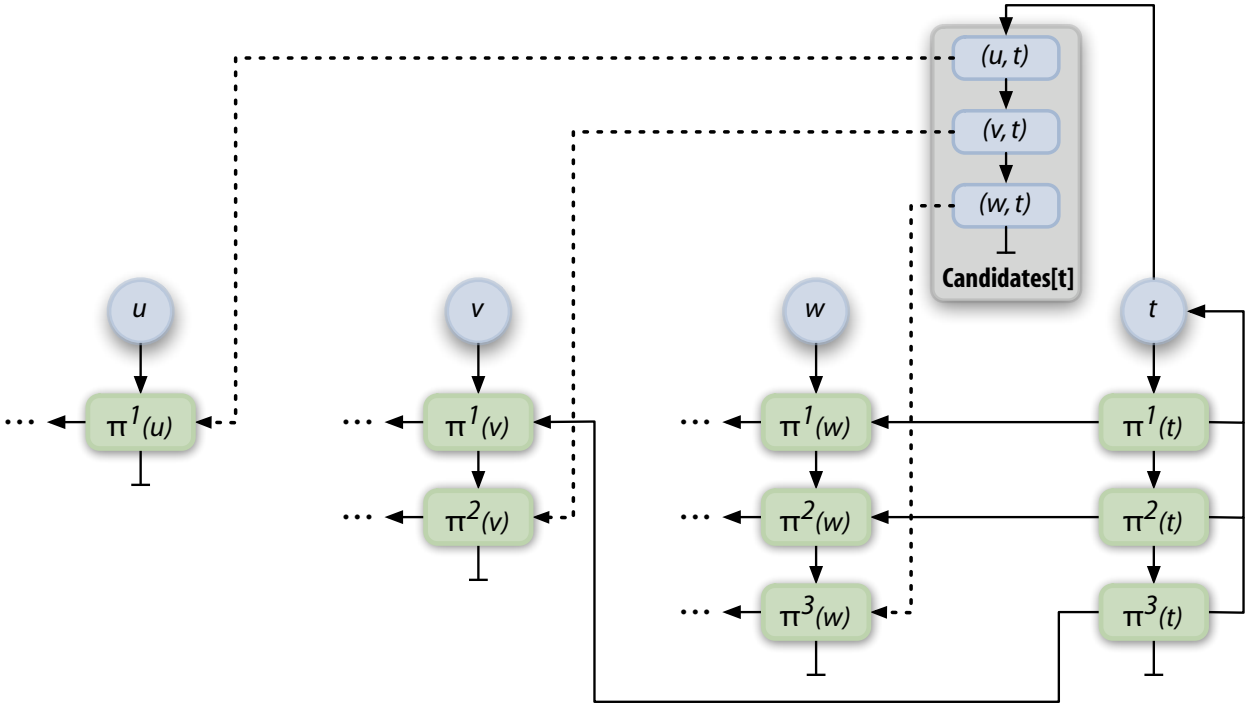
with

$$\mathbf{n}_t = (n_{t,1}, \ldots, n_{t,M})', \qquad \mathbf{p}_t = (p_{t,1}, \ldots, p_{t,M})',$$

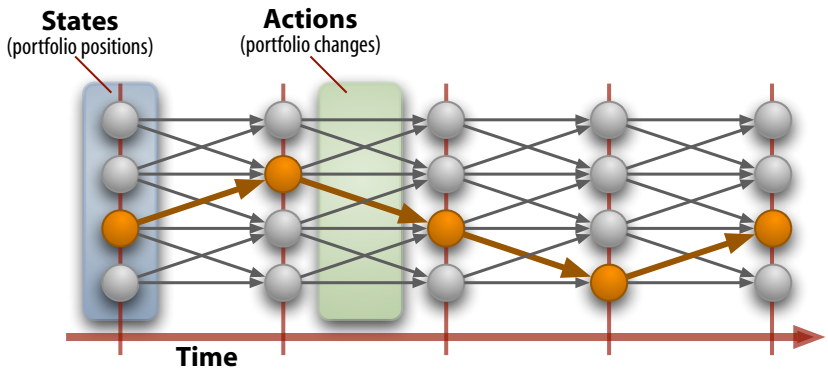
where $n_{t,i} \in \mathbb{Z}$ is the number of shares of asset i held at time t, and $p_{t,i} \in \mathbb{R}_+$ is the price of asset i at time t. We can only hold an integral number of shares and short (negative) positions are allowed. The possible actions are $\mathbf{u}_t \in \mathbb{Z}^M$ which are interpreted as buying or selling the number $u_{t,i}$ of shares asset i. To limit the search space, both $n_{t,i}$ and $u_{t,i}$ may be restricted to a small integer. A cartoon illustration of this formulation is shown in Fig. 5.12.

The (stationary) state-transition function is thus written as

$$\mathbf{x}_{t+1} = f(\mathbf{x}_t, \mathbf{u}_t, \mathbf{r}_{t+1}) = \begin{bmatrix} \mathbf{n}_t + \mathbf{u}_t \\ \mathbf{p}_t + \mathbf{r}_{t+1} \end{bmatrix}, \tag{5.14}$$



▲ **Figure 5.11.** *Data structure (fragment) for enumerating the K best paths ending at vertex t in a graph where vertices $u, v, w \in \Gamma^{-1}(t)$. The figure shows the set of possible candidates for the fourth best path ending at t, given that the first three best paths have been extracted. Except for t, pointers from path structures to vertex structures are omitted for clarity.*



◀ **Figure 5.12.** *Formulation of the portfolio management problem as a shortest-path problem in a graph. The states at a given time-step represent the possible portfolio positions. State transitions represent buy/sell actions on each asset.*

where $\mathbf{r}_{t+1} \equiv \mathbf{p}_{t+1} - \mathbf{p}_t$ is the vector of (absolute) asset returns between times t and $t+1$. It should be noted that, by the hypothesis of "no market impact", the price portion of the state is uncontrolled: it evolves independently of the action, and is therefore shared by all portfolios at a given time.

We now turn to a consideration of various source and target utility functions.

5.4.1 The `MinCost` Source Utility

As a first example of cost structure (see §5.2/p. 133), one may let the immediate cost function $g_t(\mathbf{x}_t, \mathbf{u}_t)$ at time t be the monetary (e.g. \$) amount required to carry out the decision $\mathbf{u}_t$ (i.e. establish the desired position change), accounting for transaction costs,

$$g_t = \mathbf{p}_t' \mathbf{u}_t + \chi(\mathbf{u}_t, \mathbf{p}_t), \tag{5.15}$$

where $\chi(\mathbf{u}_t, \mathbf{p}_t)$ are the transaction costs (for clarity in what follows, we shall drop the explicit functional dependence and simply write χ_t for $\chi(\mathbf{u}_t, \mathbf{p}_t)$). This corresponds to the monetary cost of buying or selling the quantity $\mathbf{u}_t$ at price $\mathbf{p}_t$. We can then take a *source utility* function U over all time steps as the sum of negative individual costs,

$$U(g_0, \ldots, g_{T-1}) = -\sum_{t=0}^{T-1} g_t. \tag{5.16}$$

Moreover, we impose the constraints that both the initial and final portfolios be empty (i.e. they cannot hold any shares of any asset). With those constraints in place, maximizing U over a time horizon $t = 0, \ldots, T$ is equivalent to finding a strategy that *maximizes the terminal wealth* of the investor over the horizon. We call this source utility function the `MinCost` utility (since it corresponds to the path of minimum \$ cost through the state-action graph).

Note that with this utility function, we never need to explicitly represent the cash amount on hand (i.e. it is not part of the state variables) since we use the **value function itself** (*viz.* $J_t^*(x_t)$ in eq. (5.8)) to stand for the cash currently on hand. This formulation has the advantage that we never need to discretize the cash currently being held, which allows minute price variations and small transaction costs (both fixed and proportional) to be handled without loss of precision.

Proposition 10 *For an action set that is at least as large as the maximum number of shares in the portfolio and bounded transaction costs, the value of state $\mathbf{x}_t$ can be lower-bounded by*

$$J(\mathbf{x}_t) \geq \mathbf{p}_t' \mathbf{n}_t + \chi_t \tag{5.17}$$

under the `MinCost` *utility, where $\chi_t \geq 0$ depends on transaction costs only.*

Proof We proceed by backward induction from time $T-1$. At $t = T-1$ we must take action $\mathbf{u}_{T-1} = -\mathbf{n}_{T-1}$ in order to meet the null terminal portfolio constraint $\mathbf{n}_T = 0$. This action costs

$$g_{T-1} = -\mathbf{p}'_{T-1}\mathbf{n}_{T-1} + \chi(\mathbf{u}_{T-1}, \mathbf{p}_{T-1}),$$

resulting in a value at time $T-1$ (*cf.* eq. (5.16)) of

$$J(\mathbf{x}_{T-1}) = \mathbf{p}'_{T-1}\mathbf{n}_{T-1} + \chi_{T-1},$$

thereby establishing the induction base case. For $t < T-1$, we assume that eq. (5.17) holds for $t+1$. From Bellman's equation, we can write

$$\begin{aligned} J(\mathbf{x}_t) &= \max_{\mathbf{u}_t} \Big[-g_t + J(\mathbf{x}_{t+1}) \Big] \\ &= \max_{\mathbf{u}_t} \Big[\mathbf{p}'_t\mathbf{u}_t + \chi(\mathbf{u}_t, \mathbf{p}_t) + \mathbf{p}'_{t+1}(\mathbf{n}_t + \mathbf{u}_t) + \chi_{t+1} \Big] \\ &= \mathbf{p}'_{t+1}\mathbf{n}_t + \chi_{t+1} + \max_{\mathbf{u}_t} \Big[\mathbf{r}'_{t+1}\mathbf{u}_t + \chi(\mathbf{u}_t, \mathbf{p}_t) \Big] \\ &= (\mathbf{p}_t + \mathbf{r}_{t+1})'\mathbf{n}_t + \chi_{t+1} + \max_{\mathbf{u}_t} \Big[\mathbf{r}'_{t+1}\mathbf{u}_t + \chi(\mathbf{u}_t, \mathbf{p}_t) \Big] \qquad (5.18) \\ &\geq \mathbf{p}'_t\mathbf{n}_t + \chi_t, \end{aligned}$$

where the second step results from substituting eq. (5.14), (5.15) and (5.17), and the third step from taking out of the maximization the terms that do not depend on $\mathbf{u}_t$. The last step follows if we assume that $\max_{\mathbf{u}_t} \mathbf{u}_t \geq \mathbf{n}_t$ (which states that the magnitude of allowed actions should be high enough to hedge adverse upcoming asset returns), and bounded costs which can be made independent of $\mathbf{u}_t$ and absorbed into χ_t. This completes the inductive step. ■

Other source utility functions can obviously be considered, as long as they allow a solution by dynamic programming. A notable case is the log utility (applied to the relative returns net of transaction costs) which incorporates a measure of risk aversion (§2.1.4/p. 14). In section 5.4.4 we examine an approximation to the Sharpe ratio that yields good practical performance.

5.4.2 Target Utilities

Denote by $\upsilon_t = \mathbf{n}'_t\mathbf{p}_t$ the *portfolio value* at time t, and by

$$\rho_t = \frac{\upsilon_t - \upsilon_{t-1}}{\upsilon_{t-1}}$$

the *portfolio relative return* between time-steps $t-1$ and t.

In the experiments below, we consider two target utility functions:

1. The *Average Return Per Time-Step*:

$$\bar{\rho}_T = \frac{1}{T}\sum_{t=1}^{T} \rho_t \,.$$

2. The *Sharpe Ratio*:

$$SR_T = \frac{\bar{\rho}_T - r_f}{\hat{\sigma}_T},$$

where r_f is an average *government risk-free rate* over the horizon, and $\hat{\sigma}_T$ is the sample standard deviation of returns

$$\hat{\sigma}_T^2 = \frac{1}{T-1}\sum_{t=1}^{T}(\rho_t - \bar{\rho}_T)^2 .$$

5.4.3 Choosing a Good K

The question remains of choosing an appropriate value of K for a particular problem. Assume that, given a random trajectory i, a source utility function U and a target utility function V, the utility values of the trajectory follow a joint probability distribution $p(u, v)$, where $u = U(i)$ and $v = V(i)$. This is illustrated in Figure 5.13. Assume further that we are interested in sampling trajectories that have at least an unconditional target utility of α or better, namely $v \geq \alpha$. Given an observed value of the source utility u, the probability that the target utility be greater than this level is given by

$$P(v \geq \alpha | u) = \frac{1}{p(u)} \int_{\alpha}^{\infty} p(u, \tilde{v})\, d\tilde{v} ,$$

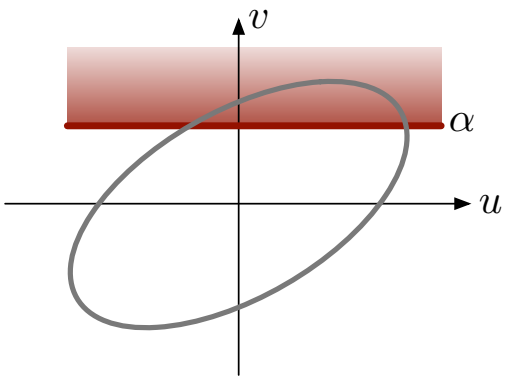
where $p(u) = \int_{-\infty}^{\infty} p(u, \tilde{v})\, d\tilde{v}$ is a normalization factor. For each trajectory i, call this probability p_i^{α}. For K trajectories, the probability that *at least one* exceeds α under V can be approximated analytically (assuming independent draws). Hence, assuming an estimator of the joint distribution $p(u, v)$, we can compute the number K that would yield a desired confidence of exceeding the target utility threshold α.

In practice, of course, the successive trajectories extracted by the K-best paths algorithm are highly dependent, since they will differ in only a single vertex; we will see in §5.4.6/p. 154 that a few thousands trajectories are enough in many cases.

5.4.4 Incremental Sharpe Ratio Source Utility

The previous analysis suggests that the higher the correlation between the source and target utilities, the quicker we should expect to find good rescored trajectories. Unfortunately, assuming the Sharpe ratio target utility, the `MinCost` source utility (which corresponds, as mentioned above to maximizing the terminal wealth) is not perfect. Although a small-but-significant positive correlation between the two is observed in practice, the lack of explicit risk aversion* in the source utility causes some "impedance mismatch" during the search.

*Apart from maximum position size and maximum allowed position change at each time-step.



◀ **Figure 5.13.** *Exploiting the correlation between source and target utilities: the number K of extracted paths should be large enough to adequately sample the "good target utility" region (shaded).*

Inspired by Moody and Saffel's *differential Sharpe ratio* (Moody and Saffell 2001b), we attempt to define a source utility function that would be more correlated with a Sharpe ratio (assuming this is the desired target). We define the *incremental Sharpe ratio* as:

$$ISR_t = \frac{\tilde{\rho}_t}{\tilde{\sigma}_t}$$

where $\tilde{\rho}_t$ and $\tilde{\sigma}_t$ are, respectively, exponentially-weighted moving average (EWMA) estimators of the return and volatility, using α as the EWMA decay factor:*

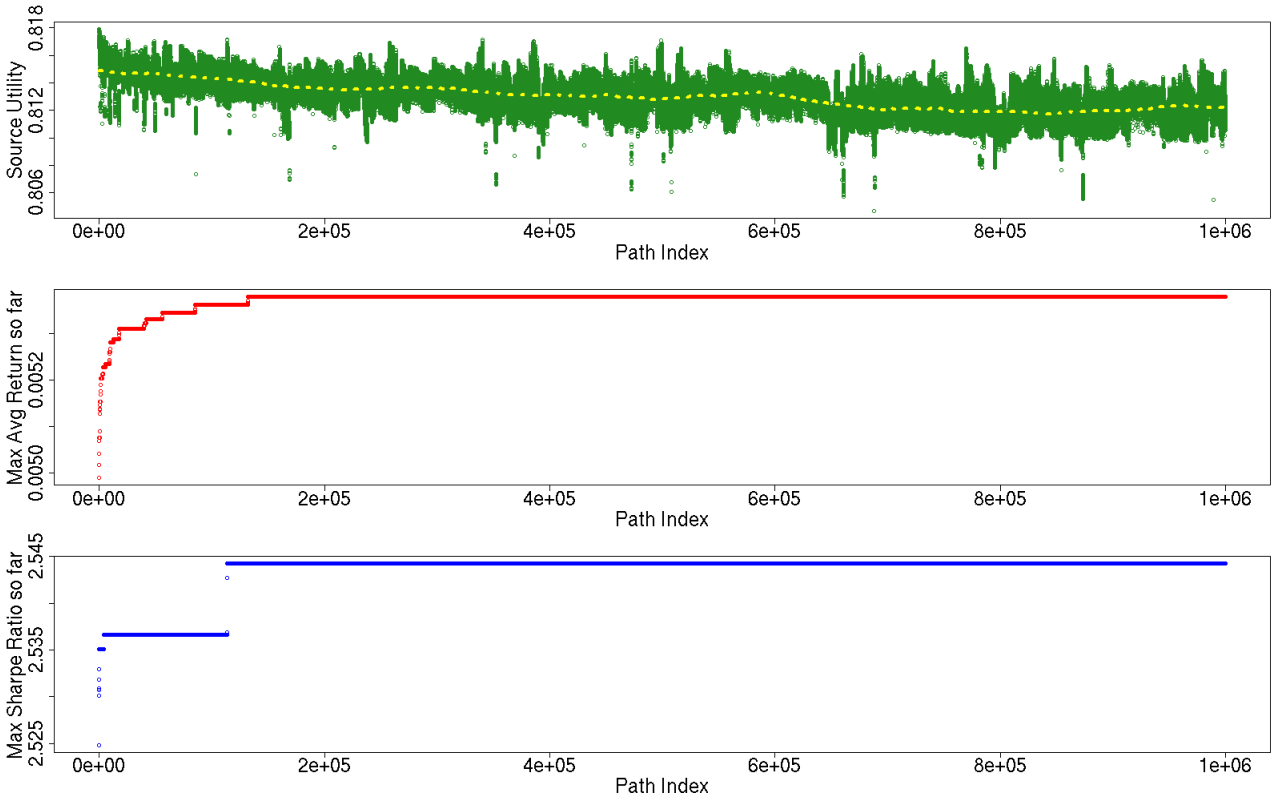
$$\begin{aligned} \tilde{\rho}_t &= \alpha\tilde{\rho}_{t-1} + (1-\alpha)\rho_t\,, \\ \tilde{\sigma}_t^2 &= \alpha\tilde{\sigma}_{t-1}^2 + (1-\alpha)\rho_t^2\,. \end{aligned}$$

These estimators are kept for each state and are updated using portfolio returns ρ_t (net of transaction costs) that are encountered along the trajectory, as explained next. At first sight, it appears that keeping such running $\tilde{\rho}_t$ and $\tilde{\sigma}_t^2$ estimators in the search would increase the state space size by two dimensions. However, a trick can be applied wherein those estimators are kept as the (compound) *value of the state*, rather than be added to the state space. In other words, when considering the Bellman equations of eqq. (5.7) and (5.8), we compute the "effective value" $J_t(x_t)$ of a state x_t as

$$J_t(x_t) = \frac{\tilde{\rho}_t(x_t)}{\tilde{\sigma}_t(x_t)}. \tag{5.19}$$

During the K-best search, we simply update the pair $\langle\tilde{\rho}_t(x_t), \tilde{\sigma}_t^2(x_t)\rangle$ as though it was a single value in the update equation (5.8). This allows to perform an approximate search according to a Sharpe-like criterion at no additional cost in terms of state space size.

*In our experiments, we used $\alpha = 0.94$ as the decay factor, which is the RiskMetrics-recommended standard value for daily data (RiskMetrics 1996).

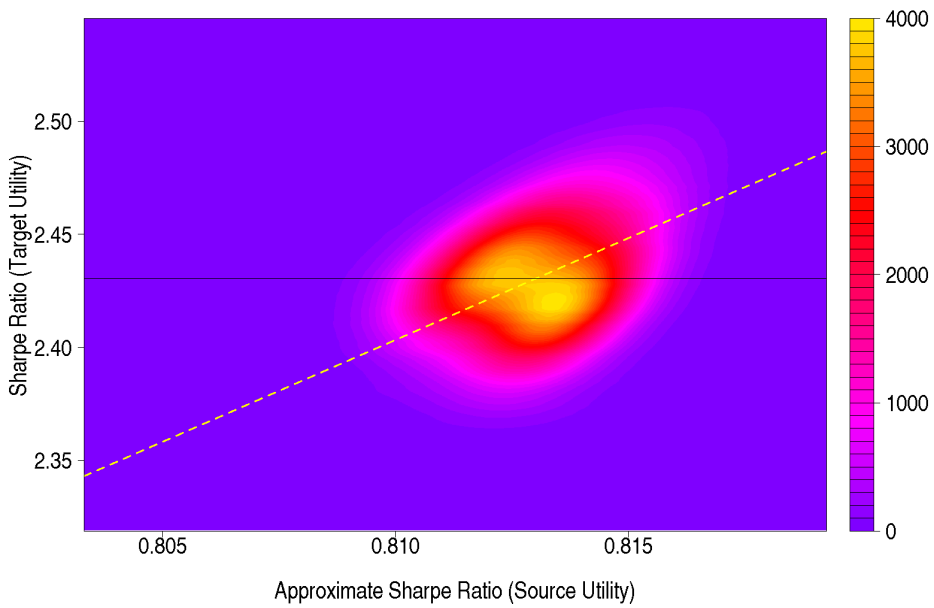


▲ **Figure 5.14.** *Utility as a function of the extracted path index, in the order found by the K-best-paths algorithm ($K = 1 \times 10^6$).* ***(Top)*** *The source utility function (incremental Sharpe Ratio), and a smoothed version thereof (dashed line); the* average *source utility decreases slowly as a function of the path index.* ***(Middle)*** *First target utility: average return per time-step (running maximum value).* ***(Bottom)*** *Second target utility: Sharpe Ratio (running maximum value).*

One could legitimately wonder whether this procedure has any justification. Indeed, using eq. (5.19) to define the value of a state **yields a non-additive cost structure**. In other words, the "optimal" trajectory implied by eqq. (5.7) and (5.8) is no longer guaranteed to be the best one. Due to the non-additivity of the incremental Sharpe ratio criterion, the K extracted paths will not exhibit a monotonic decrease in utility (as would be normal under an additive criterion); rather, the extracted path utilities will be "noisy", but we should expect a decrease in the *mean path utility* as a function of the extracted path index.

To illustrate the "noisy" nature of incremental Sharpe ratio criterion, Figure 5.14 shows various utility functions as a function of the index of the K-th best path, when extracting 1.0×10^6 paths from a historical price sample.* First, we observe that despite the noisy nature of the source utility as

*This was run on a four-asset problem (futures on British Pound, Sugar, Silver, Heating Oil) where we allow from -3 to $+3$ shares of each asset in the portfolio; maximum vari-



◀ **Figure 5.15.** *Kernel density estimate of the relationship between the incremental Sharpe ratio (source utility) and the Sharpe ratio (target utility) for a typical trajectory; the yellow dashed regression line underscores the strong relationship between the source and target utilities.*

a function of the path index, the *mean source utility* decreases, as indicated by the dashed line on the top panel (resulting from a kernel-smoothing estimator the utility). Second, we note that fairly quickly, the quality of the rescored trajectories stops increasing for both the *average return per time-step* and *Sharpe ratio* target utilities. This suggests that the source utility function captures well the traits that are required of a successful trajectory under the target utilities.

Figure 5.15 illustrates this idea. It shows a kernel density estimate (Wand and Jones 1995) of the joint distribution between U (terminal wealth utility) and V (Sharpe ratio utility), along with the regression line between the two (dashed blue line), for the same sample path history as reported in Figure 5.14. We can also compute the relative merits of the Terminal Wealth versus Incremental Sharpe ratio source utilities, in terms of their correlation with Average Return per time-step and Sharpe ratio target utilities. Using again the same sample trajectory as previously, we obtain the following correlation structure between the source and target utilities:

	Source Utility	
	Terminal Wealth	Incr. Sharpe Ratio
Avg. Return	0.25	−0.20
Sharpe Ratio	0.01	0.45

ation of +1 or −1 share per time-step; proportional transaction costs of 0.5%, trajectory length = 30 time-steps).

5.4.5 Making the Search More Efficient

A necessary step before carrying out the K-best path search with the REA procedure is to solve the single-source shortest-path problem for all portfolio states. Even when using sample trajectories, this search is subject to the curse of dimensionality due to:

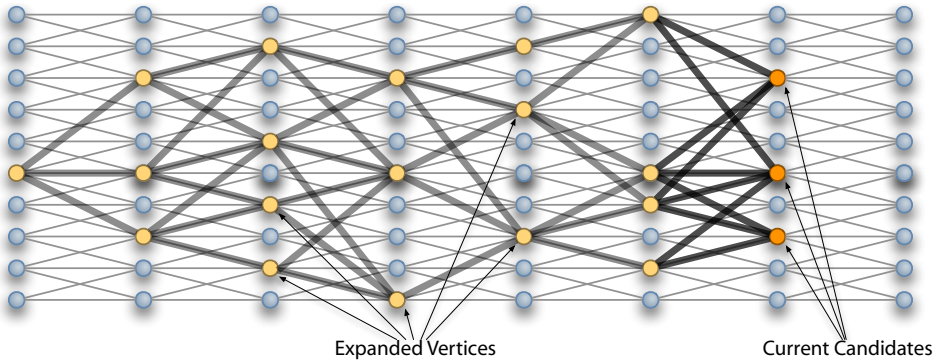
- *Size of the state space*: according to the problem formulation given in §5.4/p. 142, this grows as $O(N^M)$, where M is the number of assets and N is the maximum position size (more precisely, twice the "maximum absolute number of increments" as defined below).
- *Size of the action space*: this quantity is also exponential in the number of transitions from one time-step to the next.

The key to managing state space explosion is to only explore the most promising portions of the state–action graph. In particular, we make use of a *beam searching* procedure to avoid explicitly representing all states, and an efficient enumeration of the actions to consider only the state transitions that are most likely to be effective.

The search itself through the position/action graph, depicted in Fig. 5.12, is controlled by a number of parameters. In particular, we denote the minimum representable position in an asset by an *increment*. The following parameters govern the difficulty of the search:

1. `MAXINCR`: The *maximum absolute number of increments* that can be assumed in a state (i.e. how large can positions be). For simplicity, this is assumed to be the same in the positive and negative directions. This directly affects the size of the state space: at each time-step, there are $(2 \times \texttt{MAXINCR} + 1)^M$ possible portfolios (the $+1$ takes care of a position of 0 in an asset), where M is the number of assets.
2. `MAXDELTA`: The maximum increment delta that is possible for state-to-state transitions. For example, if this value is 2, the possible increments can be $\{-2, -1, 0, 1, 2\}$. Obviously, this set is restricted if it would take the position state outside the range allowed by `MAXINCR`. This parameter directly affects the size of the action space: at each time-step, there are at most $(2 \times \texttt{MAXDELTA} + 1)^M$ possible transitions to the next time-step.
3. `GRANULARITY`: This is a divisor that maps the number of increments into a number of physical financial contracts (or number of shares); for instance, if a state position (in units of increments) has a value of 8 and the granularity is 4, this corresponds to taking a physical position size of 2 contracts.

Collectively, we shall refer to these parameters as the *density*, specified as a triple (`MAXINCR`, `MAXDELTA`, `GRANULARITY`).



◀ **Figure 5.16.** *Illustration of the beam search applied to the state-action graph. Only some of state transitions are illustrated for clarity. The expanded vertices, linked by thick edges, are explicitly represented in memory; the others may have been considered, but then fell out of the beam.*

Beam Search

A beam search (Russell and Norvig 2002) is used to represent only the most promising portions of the state-action graph. We implement the search sequentially, one time-step at a time. At each time t, *candidate vertices* at time $t+1$ are expanded (based on an efficient enumeration of the possible actions; see next). Only the highest-valued candidates are kept at each time. The approximation to the search arises since some candidates are discarded, but the overall best path through the graph could pass through one of the discarded candidate; the overall "best" number of candidates to keep at each stage results from a trade-off between search accuracy and resource consumption.

A depiction of the procedure is given in Fig. 5.16. The vertices actually visited by the search are linked by thick edges: only those are actually represented in memory; all the others fell out of the beam and are only implicitly part of the graph. The candidates at time $t+1$ can only emerge from the set of expanded vertices at time t.

Since the beam search proceeds in the forward direction (for $t = 1, \ldots, T-1$), the value associated with each vertex is the cost-so-far, namely the value of the source utility from the starting vertex (initial portfolio) to the current one.

As for A* search (Hart, Nilsson, and Raphael 1968), beam search requires an approximation of the **total cost** through the graph of a path passing through a given vertex. When searching in the forward direction, this can be given as the sum of the exact cost-so-far and a *heuristic* approximating the cost-to-go. For the `MinCost` source utility, a natural heuristic is given by the bound on the cost-to-go established by proposition 10.*

The result the beam search provides the "skeleton" over which the K-best paths search can be performed: in our implementation, only the set of expanded vertices in the beam search are allowed to be part of solutions returned by the REA procedure. The beam-width parameter allows, in effect,

*For the `IncrSharpe` source utility, the heuristic used is given by the constant 'zero' for simplicity, although little effort has been paid to finding a good heuristic in this case.

control over the memory and time consumption of the complete procedure.

Efficient Enumeration of the Action Set

Another key to alleviating the curse of dimensionality is to efficiently enumerate the actions to transition from one time-step to the next. This is possible in the case of the `MinCost` source utility. Equation (5.18) expresses the value of a state $\mathbf{x}_t$ as a function of the current number of shares $\mathbf{n}_t$ and action taken at time t, $\mathbf{u}_t$. Neglecting transaction costs, we observe that the value is maximized if the quantity

$$\max_{\mathbf{u}_t} \left[\mathbf{r}'_{t+1} \mathbf{u}_t \right]$$

is maximized, leading the way to efficiently enumerating actions: the best action is seen to have the *largest dot product* with the asset return vector $\mathbf{r}_{t+1}$. Hence, by enumerating the actions $\mathbf{u}_t$ in **order of decreasing dot product** with $\mathbf{r}_{t+1}$ and maximizing the quantity

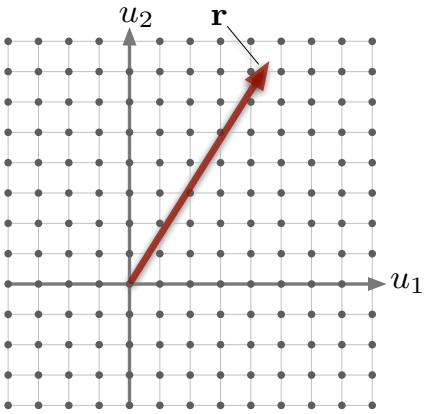
$$\mathbf{r}'_{t+1} \mathbf{u}_t - \chi(\mathbf{u}_t, \mathbf{p}_t) \tag{5.20}$$

to account for transaction costs, we can achieve significant pruning in the action enumeration. Moreover, at a given time t, the order of the actions is **independent** of the number of shares $\mathbf{n}_t$, depending only on the uncontrolled portion of the state. Hence, the action enumeration order can be precomputed for all timesteps, giving rise to a significant further speedup.

The actions are discretized whereas the asset returns form a real vector; the situation is illustrated in Fig. 5.17. We shall denote by $\mathbf{r} \in \mathbb{R}^M$ the "reference" vector with respect to which we want to compute the dot product, and by $\mathcal{U} \subset \mathbb{Z}^M$ the finite set of vectors that we wish to enumerate by decreasing of dot product with $\mathbf{r}$. Without loss of generality, we shall assume that elements of $\mathcal{U}$ are sampled on a regular grid ranging from N^- to N^+ along each dimension (assumed finite), with a fixed interval Δ between each element. We also assume that the elements of $\mathbf{r}$ are all strictly positive. (Negative elements can be handled and treated as positive with the provision that actions must be enumerated in the *opposite order*, from N^- to N^+ instead of from N^+ to N^-, in the algorithm below.)

The algorithm to enumerate elements of $\mathcal{U}$ in decreasing dot-product order with $\mathbf{r}$ is shown in Listing 5.3. It relies on a priority queue data structure (Cormen, Leiserson, Rivest, and Stein 2001) supporting two operations:

- $\mathsf{Push}(P, x, c)$ adds element x to the priority queue P with associated cost c.
- $(x, c) \longleftarrow \mathsf{Pop}(P)$ removes the element x having the currently-highest cost c from the priority queue.



◀ **Figure 5.17.** *For the* `MinCost` *source utility, the actions (here* $\mathbf{u} = (u1, u2)$*) can be efficiently enumerated in order of decreasing dot product with the return vector* $\mathbf{r}$*, forming the basis for search pruning. Each black dot represents a possible action choice for* $\mathbf{u}$*.*

We assume that the `yield` statement in the pseudocode is used to return iterable elements.*

Listing 5.3: Dot-Product Order Enumeration Algorithm

```
def DotProductOrderEnumeration(r, Δ, N⁻, N⁺):
    # Initialize the queue with a vector of length M at maximum dot−product
    P ⟵ NewPriorityQueue()
    u ⟵ (N⁺, ..., N⁺)
    Push(P, u, u'r)

    # Retrieve each element from the queue in turn
    while not Empty(P):
        (u, c) ⟵ Pop(P)

        # Push new elements arising from updating dot product
        for i = 1, ..., M:
            if u_i ≥ N⁻:
                c̃ ⟵ c − r_i Δ
                ũ ⟵ u
                ũ_i ⟵ ũ_i − Δ
                Push(P, ũ, c̃)

        yield u
```

From inspection, the algorithm terminates since each element of $\mathcal{U}$ is pushed to the queue only once, and $\mathcal{U}$ is assumed to be finite by hypothesis. Queue management requires $O(\log N)$ time per push or pop for a naïve implementation (e.g. binary heaps), although a Fibonacci heap allows pushes to be made in amortized constant time. It is assumed that only a tiny fraction of $\mathcal{U}$ will need to be visited for eq. (5.20) to be satisfactorily (approximately) maximized.

*Similarly to the statement of the same name in the Python programming language.

5.4.6 Optimization by K-Best Paths

It remains to evaluate the benefits brought forth by the K-Best paths procedure, and this section investigates in detail its behavior for this purpose. We consider the optimization of single-asset decision sequences for the following major equity indices: CAC 40 (from Jan. 1991), DAX 30 (from Jan. 1991), Russell 1000 (from Jan. 2000) and S&P 500 (from Jan. 1990). All data ends in Dec. 2007. For each asset, we form all non-overlapping price sequences of respective lengths 21 days (1 month), 63 days (3 months), 126 days (6 months) and 252 days (1 year), and attempt to determine the decision sequence maximizing the Sharpe Ratio (5.4), subject to varying levels of transaction costs.

We compare the following settings:

- Optimization by conjugate gradients descent alone, from a "zero" starting point (i.e. the starting point for the optimization is a constant neutral decision sequence).

- Optimization by K-Best paths, using the "maximum return" source utility function, followed by rescoring according to the Sharpe ratio. These decisions serve as the starting point for a gradient-based "fine-tuning", proceeding as in the previous step, and that mitigates the discrete optimization performed by the K-Best paths procedure. This source utility function is called `MinCost`.

- Same as previously, but using the (noisy) Incremental Sharpe ratio source utility introduced in §5.4.4/p. 146. This is called `IncrSharpe`.

For experiments involving K-Best paths, the number of paths, K is varied from 10^0 to 10^4 by powers of ten. In the first results, we fix the search density (*cf.* §5.4.5/p. 150) to `(1,2,1)` for experiments that involve K-Best search. We consider the impact of density on performance in §5.4.6/p. 157.

Impact of K on Sharpe Ratio After Rescoring

Restricting attention to the behavior of K-Best paths procedure itself, we analyze the factors that induce variations in performance. Figure 5.18 shows the mean Sharpe ratio across all price sequences, measured for several levels of transaction costs and number of extracted paths (K), for the following cases:

- For the `MinCost` source utility, immediately after rescoring according to the Sharpe ratio but before fine-tuning (see Fig. 5.9 for details about path extraction and rescoring);

- For the `IncrSharpe` source utility, after rescoring but before fine-tuning;

- Same two source utilities as above, but after performing a "fine-tuning" by gradient descent to maximize a regularized Sharpe ratio.

In all cases, we plot the Sharpe ratio (the real objective of interest), without any of the regularization term that are present within the objective functions that are actually optimized (*cf.* eq. (5.4)).

We note that:

- The dependence on the number of extracted paths, K, is very small, and mostly present within the first 10–100 paths. There is no tangible gain in extracting a larger number of paths.

- Fine-tuning by gradient descent brings a large benefit when no transaction costs are involved, mostly since decision variables are allowed to take real-valued settings that closely adjust to the realized price sequences. However, the benefits quickly fall off as costs increase, and are completely inexistent for 100 basis points and beyond.

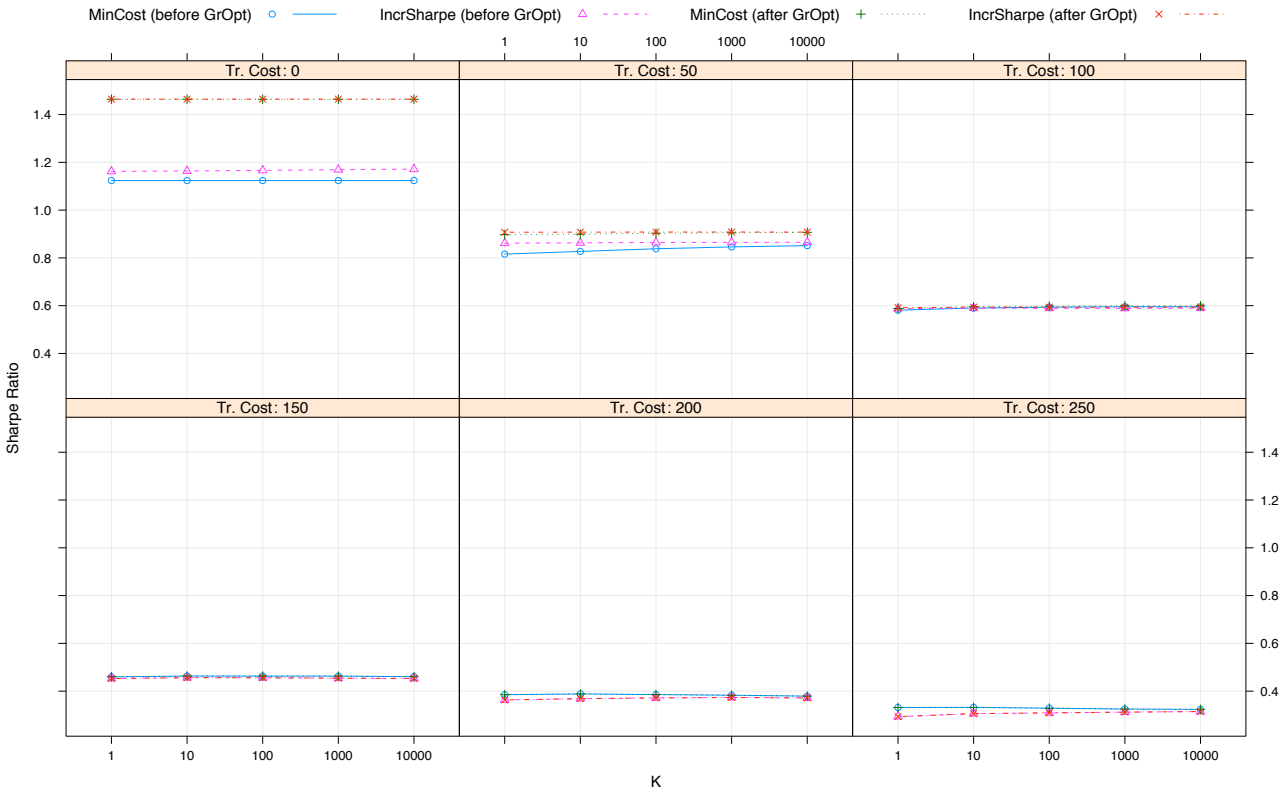
- There is no clearly "better" source utility function: at relatively low cost levels (100 BPs and below), the `IncrSharpe` function shows a small advantage, whereas the opposite is true at larger costs. This may be due to transaction costs degrading the approximation performed by the `IncrSharpe` function.

Impact of Source Utility

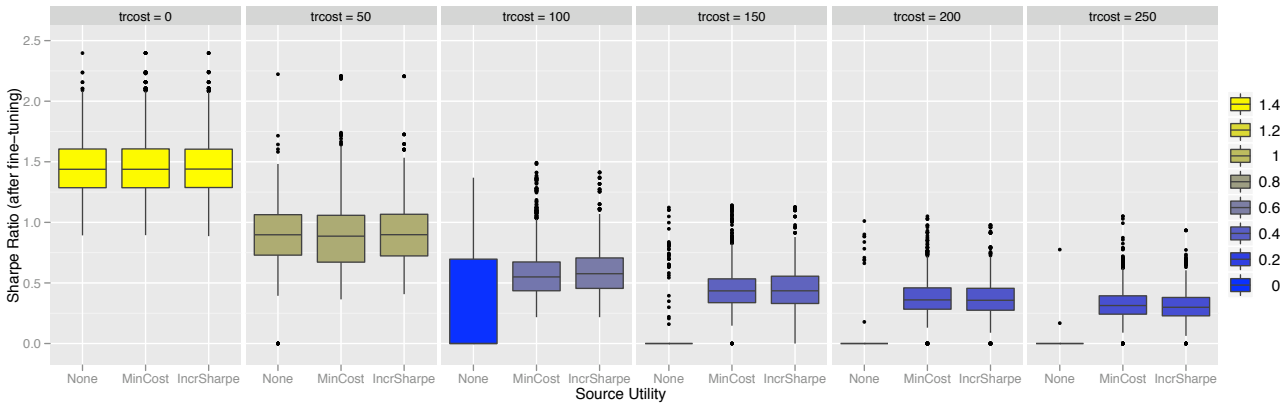
A different perspective is obtained by comparing the distribution of Sharpe ratios after fine-tuning, across transaction costs, for decision sequences obtained by pure gradient descent versus K-best paths (followed by fine-tuning by gradient descent). Figure 5.19 shows boxplots that summarize the distributions* We note that above 100 BPs, there is a catastrophic breakdown of the pure gradient-based approach, making it incapable (except in a few exceptional cases) of moving away from the all-neutral starting point. In contrast, we see that an initial optimization based on K-best paths keeps producing comparatively far better performance even at high costs.

Beyond this, there appears to be little practical difference between the `MinCost` and `IncrSharpe` source utilities.

*Price sequences of various lengths (63, 126 and 252) are pooled together in this plot, since all Sharpe ratios are annualized, making the scales comparable; there is very little difference between the behavior at those three time horizons, albeit with a variance that decreases slightly as the time horizon increases. In contrast, the behavior for 21-day price sequences differ somewhat, since from a cost level of 150 BPs, there is a disproportionate fraction of decision sequences that turn out to be precisely zero (neutral), given that 21 days is a very short horizon to invest profitably when costs are high.



▲ **Figure 5.18.** *Impact of K, the number of extracted paths, on the Sharpe Ratio objective, for two source utilities (*`MinCost` *and* `IncrSharpe`*) and various transaction costs levels, both immediately after rescoring but before fine-tuning by gradient optimization ("Before GrOpt") and after fine-tuning ("After GrOpt"). Transaction costs are measured in basis points.*



▲ **Figure 5.19.** *Impact of source utility on Sharpe ratio after fine-tuning with gradient optimization, as a function of transaction costs (in* BPs*). "None" means that only gradient optimization is performed, with no K-Best search. Beyond 100* BPs*, we note that the K-Best procedure becomes essential; with little marked difference between the* `MinCost` *or* `IncrSharpe` *source utilities apparent from the plot.*

Sequence-Level Comparisons

Whereas the previous results showed an overall distribution of Sharpe ratios, we now turn to a comparison at the level of individual price sequences. For each of the price sequences described in §5.4.6/p. 154, we compare the performance obtained under the K-Best paths procedure (after gradient-based fine-tuning) to the performance obtained by plain gradient descent. The resulting distributions of *Sharpe ratio differences* are plotted in Fig. 5.20 as a function of transaction costs, asset, price sequence length and source utility function (either `MinCost or IncrSharpe`). A positive difference indicates that the K-Best procedure yields a better performance than the alternative gradient optimization.

The results largely confirm earlier intuitions: in the complete absence of costs, there are no benefits to the K-Best procedure, and only a marginal benefit at low costs (50 BPs) which is more marked for very short sequences (21 days). The real differences start emerging from transaction cost levels of 100 BPs and above, where the consistent performance advantage of the K-Best method are clearly evident, across all price sequence lengths* and assets. The sharp increase in the *variance* of the Sharpe ratio difference at 100 BPs can be interpreted as a phase transition phenomenon: at this cost level, some price sequences remain optimizable by gradient descent, but many more start not to be; beyond this cost level, most sequences stop being optimizable by gradient descent.

A comparison of the top and bottom figures confirms that there is very little practical difference between the `MinCost` and `IncrSharpe` source utilities.

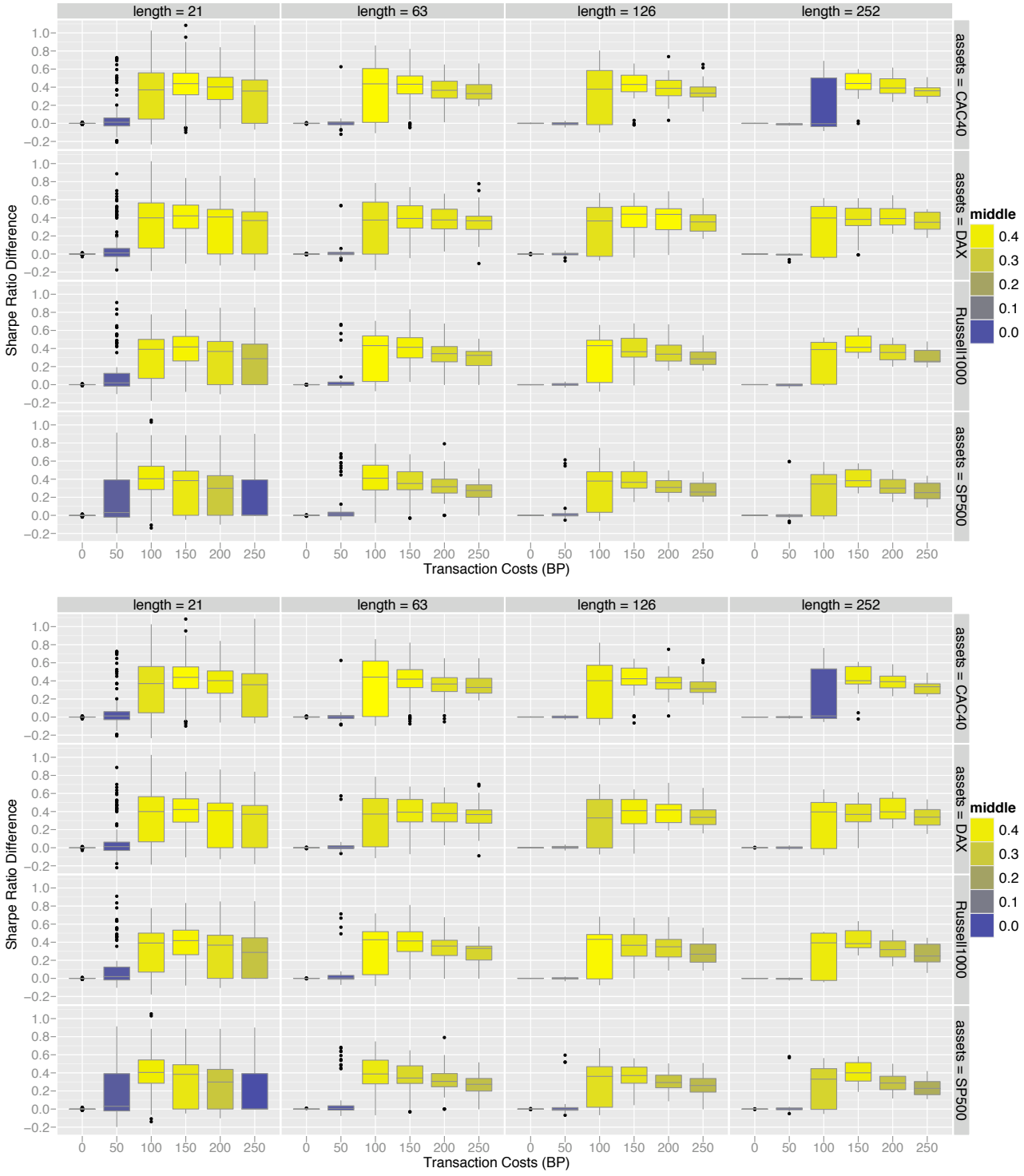
Comparing K-Best Search Parameters

Finally, we briefly analyze the effect of the K-best *search density* parameters on performance. The two settings compared are a "coarse setting" (with parameters `(1,2,1)`)† and a "medium setting" (with setting `(4,4,4)`). Both these settings can take a maximum (absolute) position of one physical contract in the traded asset, but the latter also allows fractional positions ($\frac{1}{4}, \frac{1}{2}, \frac{3}{4}$) to be taken.

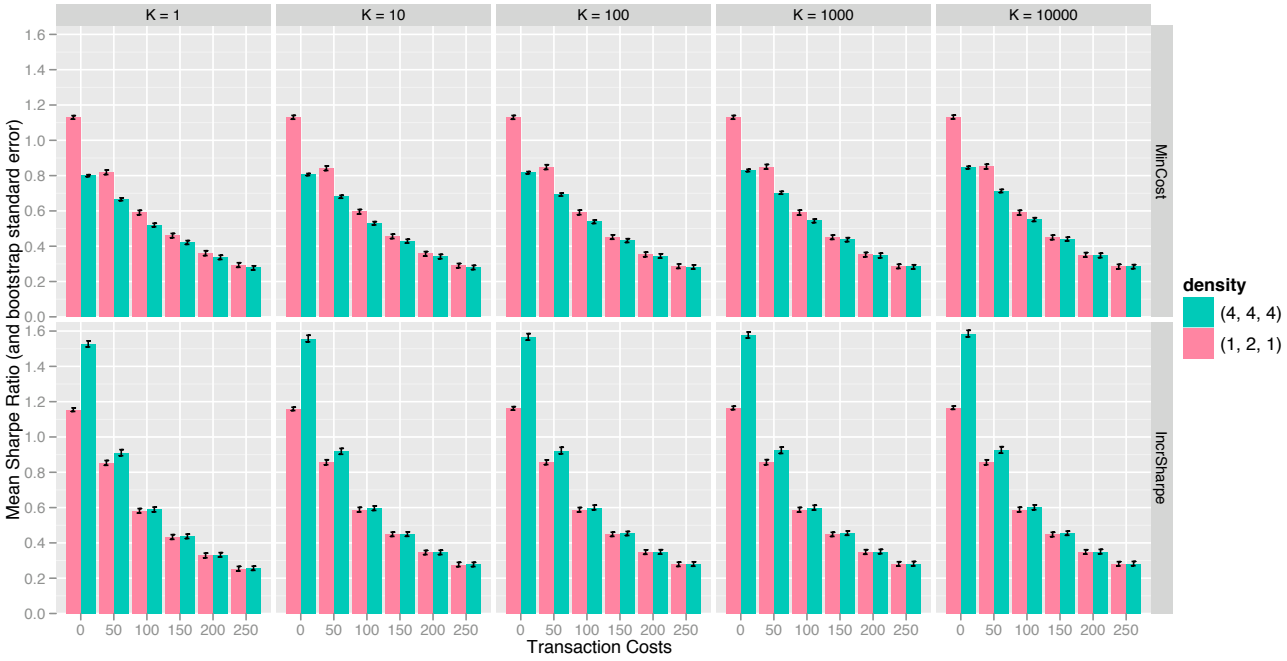
The first performance metric is the rescored *mean Sharpe ratio* after K-best search, before any gradient-based fine-tuning. Figure 5.21 shows this measure as the transaction costs, the number of extracted paths K, the density setting and the source utility are varied. The remaining experimental conditions (i.e. sequence length, underlying assets and time periods) remain as described in §5.4.6/p. 154. The error bars on performance represent one standard error on the mean, obtained by a robust bootstrap procedure.

*For 21-day sequences, it was already noted that with high costs, it may be optimal to remain neutral for the entire period; it is for this reason that the lower-end of the boxplots are around zero for sequences of this length.

†Indicating, respectively, `MAXINCR`, `MAXDELTA` and `GRANULARITY`; see §5.4.6/p. 154.



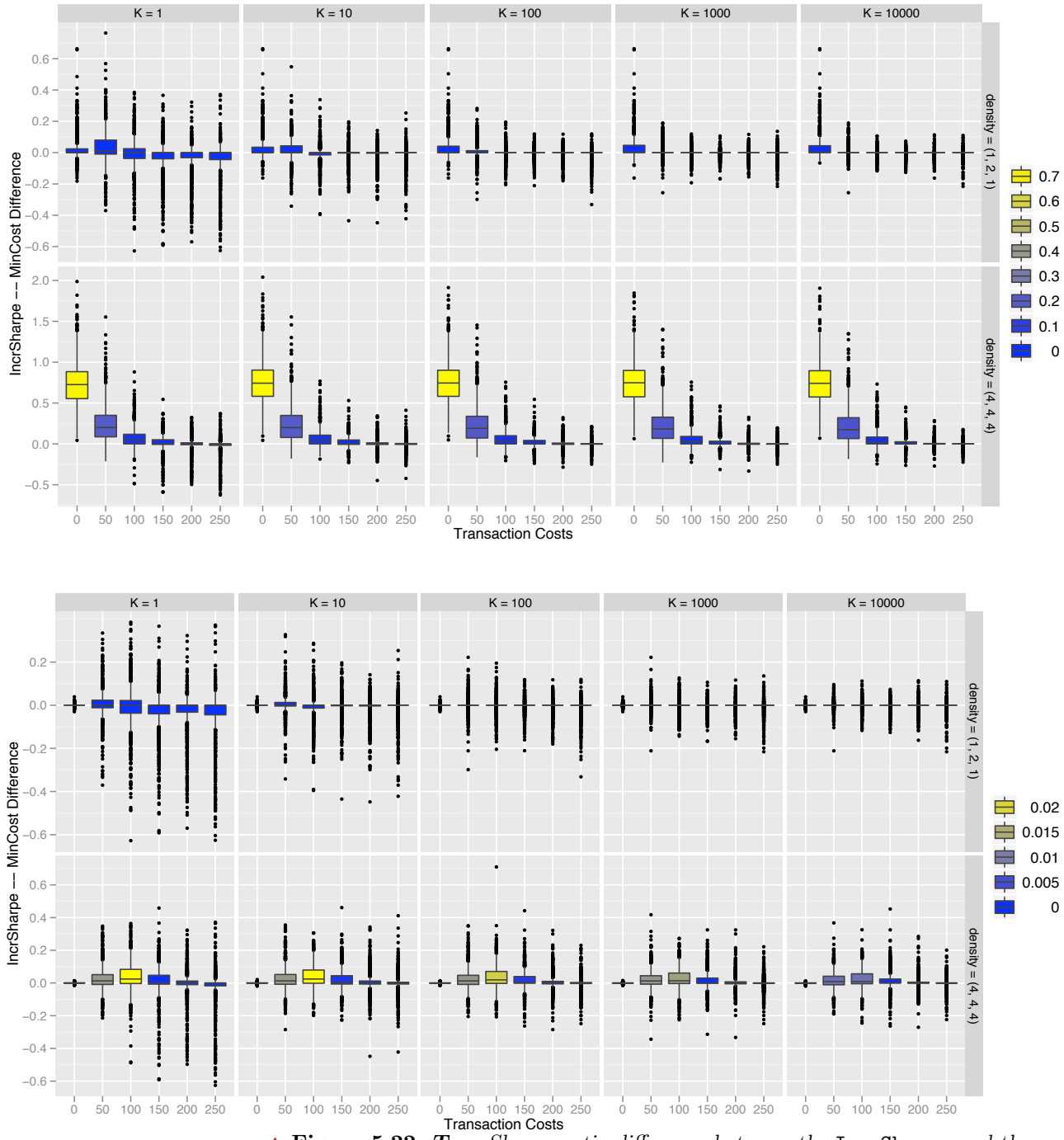
▲ **Figure 5.20.** ***Top:*** *Plot of the difference in Sharpe ratio between decision sequences optimized by the K-Best paths procedure (using the* `MinCost` *source utility) followed by gradient descent and decision sequences optimized by gradient descent alone, as a function of transaction costs, market and sequence length. Above 100* BPs*, the K-Best paths procedure always produces better median results.* ***Bottom:*** *Same experiments, using the* `IncrSharpe` *source utility. For trajectories optimized by K-Best paths,* $K = 10000$ *is used throughout.*



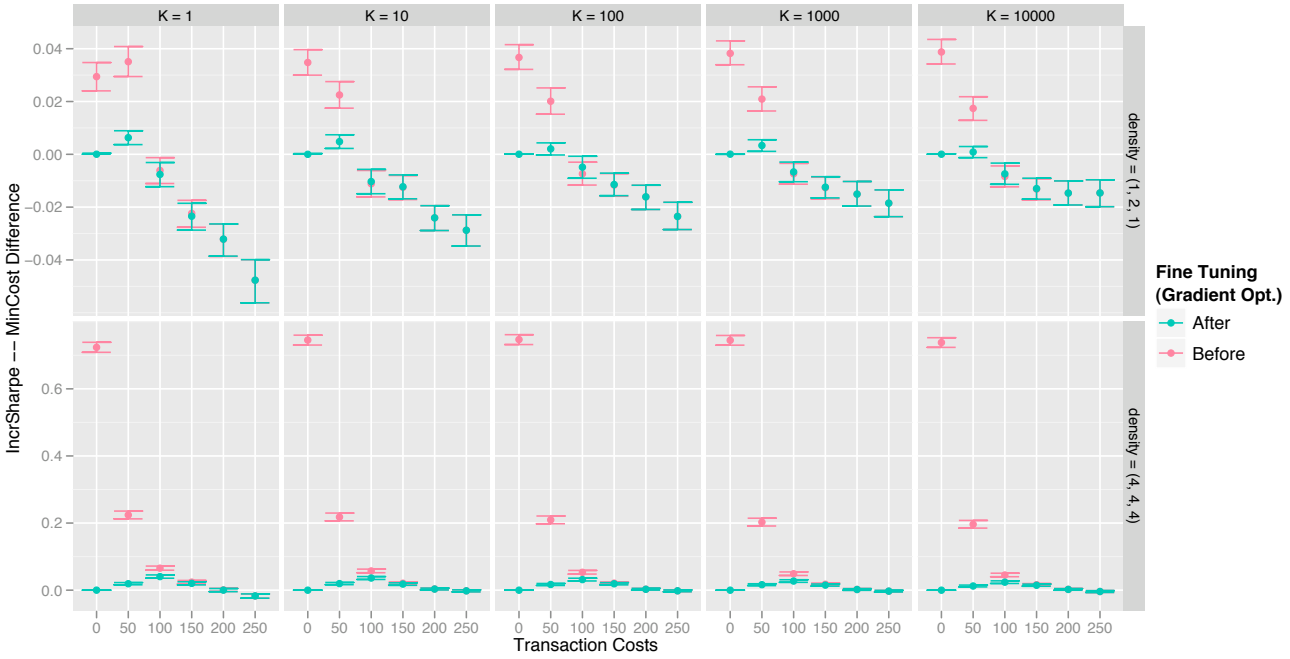
▲ **Figure 5.21.** *Mean Sharpe ratio after K-best paths rescoring, by transaction cost, the number of extracted paths K, the K-best search density and the source utility. The error bars represent one standard error on the means and are obtained by a robust bootstrap resampling procedure.*

Predictably, the obtainable Sharpe ratios decrease with transaction costs. The behavior under the density parameter is more complicated: the `IncrSharpe` utility performs considerably better at the finer setting than the coarser one, particularly at low costs, whereas the opposite is true for the `MinCost` utility. Undeniably, given the opportunity to meticulously explore the space, the `IncrSharpe` utility produces higher-yielding outcomes; but these appear to be opportunities washed away by cost-induced frictions. The dramatic underperformance of the higher-density setting with `MinCost` utility at low costs is more puzzling and seems to reflect the different nature of the source and target utilities: the action sequences obtained under the higher-density setting would maximize performance under the `MinCost` source utility, but perform badly after rescoring under the Sharpe ratio target utility. In a sense, they "overfit" the source utility, an interesting manifestation of the bias–variance trade-off (§3.1.3/p. 73).

To gain a better insight into the difference between the two source utilities, we compare them at the sequence level. Figure 5.22 shows boxplots summarizing the distribution of the *difference* between the `IncrSharpe` and `MinCost` source utilities, across transaction costs, number of extracted paths K and search density. The upper figure shows this difference after rescoring, and the bottom after gradient-based fine-tuning. The boxplots are color-coded according to the value of the median.



▲ **Figure 5.22.** ***Top:*** *Sharpe ratio difference between the* `IncrSharpe` *and the* `MinCost` *source utility functions after K-best search but before gradient-based fine-tuning, across number of extracted paths (K) and transaction cost levels, for two "search density" parameters. For the higher-density K-best search setting,* `IncrSharpe` *significantly outperforms* `MinCost`*, on average, at low to moderate transaction costs, although heavy tails are observed in the performance difference. For clarity, note that the y axes are on different scales.* ***Bottom:*** *Same experiments, but after gradient-based fine-tuning. The performance advantage in favor of* `IncrSharpe` *persists at the higher-density setting, albeit reduced in amplitude. We also note an underperformance of the* `IncrSharpe` *function at $K = 1$.*



▲ **Figure 5.23.** *Median difference between the* `IncrSharpe` *and* `MinCost` *source utilities, for various transaction costs, number of extracted paths (K) and search density, both before carrying out the gradient-based fine-tuning and after. The error bar represent 95% confidence intervals on the median.*

At low costs and for the higher-density setting, the outperformance of the `IncrSharpe` source utility is clear, highly statistically significant* and persists at all Ks; it vanishes as the transaction costs increase, although heavy two-sided tails remain. The difference is less dramatic after gradient-based fine tuning is applied. In particular, for $K = 1$ extracted paths, we now note an *underperformance* of the `IncrSharpe` utility (the median of which can be determined to be statistically significant), with a heavily negatively-skewed distribution. For higher Ks, the median difference vanishes. For ease of assessing significance, Fig. 5.23 shows a zoomed-out view of the median of the Sharpe ratio difference between the two utilities, along with 95% confidence intervals.

5.5 Ordinal Regression for Portfolio Allocation

The availability of "good" targets for portfolio allocation (obtained by the K-best paths method) pose problems in its own right: it is far from obvious

*Which can be assessed by a non-parametric test procedure, such as that of Bauer (1972).

what type of learning algorithm can best take advantage of them. We propose that supervised learning algorithms that are most suitable for this task do not directly fall in the traditional categories of regression or classification, but somewhere in-between, in a framework known as *ordinal regression*. Ordinal regression models (McCullagh 1980; McCullagh and Nelder 1989) attempt to fit measurements that are observed on a categorical *ordinal* scale: they solve a classification problem where there is a *natural ordering* among the classes, in contrast to more standard classification models which assume a *nominal* scale (i.e. no ordering structure whatsoever). The motivation for such a formulation arises when one can posit an unobserved "latent" real-valued variable, of which only a discretized version is observed. For example, one can grade product quality on a coarse letter scale, but assume that an underlying unobserved real-valued "true" product quality constitutes the driving factor.

The application of ordinal regression models to forecasting market direction should be obvious: instead of attempting to forecast a *precise return* of a market index over a time horizon (e.g. one month), a formulation through ordinal regression would instead forecast a return along a "coarsened" scale, e.g. the prototypical analyst ratings of "strong buy", "buy", "neutral", "sell", "strong sell". The justification for using such an approach would mainly be one of robustness: it is well-known that forecasting the first moment of an asset return distribution is notoriously imprecise. In contrast, a coarser model may lead to more robust decisions, or at least more successfully convey (to a human trader) the uncertainty regarding the outcome.*

Although the basic proportional-odds logit and probit models of McCullagh (1980) are well-known to economists and social scientists, there has been in recent years a revival of this literature in the field of machine learning, motivated in part by the "learning to rank" problem (e.g. Chu and Ghahramani (2005), Li and Lin (2007), Li, Burges, and Wu (2008)). In this vein, we propose to investigate a non-linear extension to McCullagh's original model, which has proved of interest in financial forecasting settings (Mathieson 1995; Mathieson 1997).

At its core, the model that we consider, called here a *Financial Neural Network* (or FinancialNNet for short), is trained to maximize a regularized financial criterion that takes into account transaction costs. However, due to the difficulty of carrying out the required optimization, elaborated upon at length in §5.1/p. 123, we make use of the "optimal targets" (under a given utility function, such as the Sharpe ratio) targets found by the K-best path procedure outlined previously to provide intermediate objectives that can guide the top-level optimization and mitigate the issue of local minima. The hope with this approach is to obtain *direct models* that are capable of

*This is consistent with recent results in the financial economics literature that note that direction-of-change forecasts can be significantly more robust than simple return forecasts (Christoffersen and Diebold 2006; Christoffersen, Diebold, Mariano, Tay, and Tse 2007).

better generalization than either models trained with "sub-optimal" targets, or with a less robust architecture.

5.5.1 Ordinal Regression with Proportional Odds

Let $Z \in \mathbb{R}$ be an unobserved variable and $Y \in \{1, \ldots, K\}$ be defined by discretizing Z according to ordered cutoff points

$$-\infty = \zeta_0 < \zeta_1 < \cdots < \zeta_K = \infty. \tag{5.21}$$

We observe $Y = k$ if and only if $\zeta_{k-1} < Z \leq \zeta_k, k = 1, \ldots, K$. The proportional odds model assumes that the cumulative distribution of Y on the logistic scale is modeled by a linear combination of input variables $\mathbf{x}$, i.e.

$$\text{logit}P(Y \leq k \,|\, \mathbf{x}) = \text{logit}P(Z \leq \zeta_k \,|\, \mathbf{x}) = \zeta_k - \eta(\mathbf{x}),$$

where $\eta(\mathbf{x}) = \boldsymbol{\theta}'\mathbf{x}$ is a linear combination of inputs and model parameters $\boldsymbol{\theta}$, and $\text{logit}(\cdot)$ is the log-odds function*

$$\text{logit}(p) = \log \frac{p}{1-p}.$$

The probability that Y belongs to a given class k is given by

$$P(Y = k \,|\, \mathbf{x}) = P(Y \leq k \,|\, \mathbf{x}) - P(Y \leq k-1 \,|\, \mathbf{x}), \qquad k = 1, \ldots, K. \tag{5.22}$$

Since the inverse of the logit function is the well-known *logistic sigmoid*

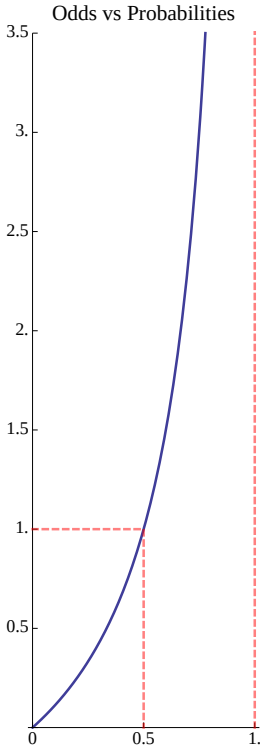
$$\text{sigm}(x) = \frac{1}{1 + \exp(-x)},$$

we have the probability of belonging to a given class $k, k = 1, \ldots, K$, as

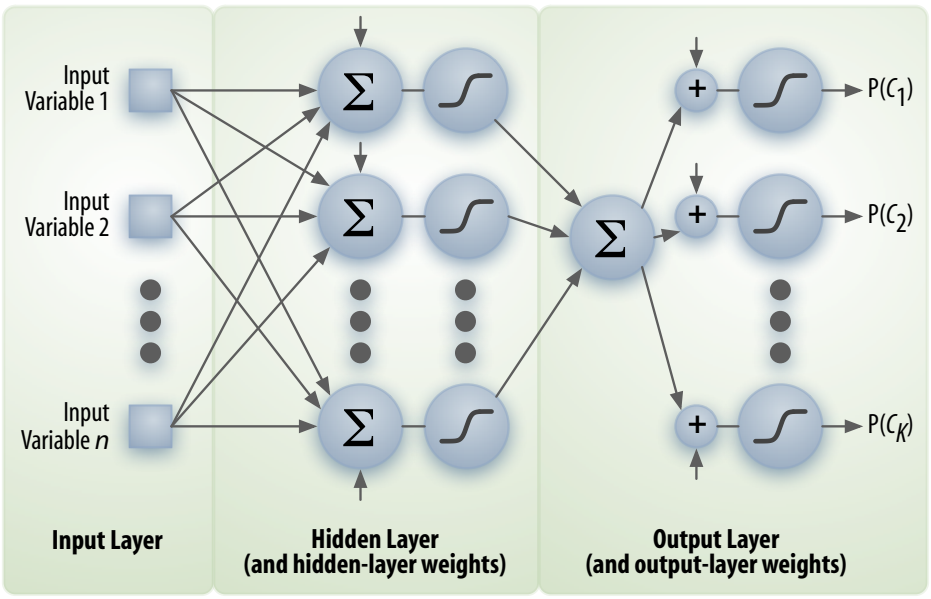
$$\begin{aligned} P(Y = k \,|\, \mathbf{x}) &= \text{sigm}(\zeta_k - \eta(\mathbf{x})) - \text{sigm}(\zeta_{k-1} - \eta(\mathbf{x})) \\ &= \frac{1}{1 + \exp(\eta(\mathbf{x}) - \zeta_k)} - \frac{1}{1 + \exp(\eta(\mathbf{x}) - \zeta_{k-1})}. \end{aligned}$$

The model is called "proportional odds" since the odds ratio between two cases $\mathbf{x}_1$ and $\mathbf{x}_2$ is independent of the class k, being only a function of the difference $\mathbf{x}_2 - \mathbf{x}_1$,

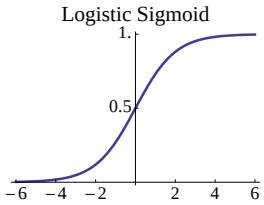
$$\begin{aligned} \frac{P(Y \leq k \,|\, \mathbf{x}_1)/(1 - P(Y \leq k \,|\, \mathbf{x}_1))}{P(Y \leq k \,|\, \mathbf{x}_2)/(1 - P(Y \leq k \,|\, \mathbf{x}_2))} &= \frac{\exp(\zeta_k)\exp(-\boldsymbol{\theta}'\mathbf{x}_1)}{\exp(\zeta_k)\exp(-\boldsymbol{\theta}'\mathbf{x}_2)} \\ &= \exp(\boldsymbol{\theta}'(\mathbf{x}_2 - \mathbf{x}_1)), \qquad k = 1, \ldots, K. \end{aligned}$$



* In statistics, the *odds* of an event having probability p are the quantity $\frac{p}{1-p}$. This quantity is sometimes more logical to work with than raw probabilities since it spans the non-negative real line. Hence, it is natural to speak of the *ratio* between the odds of two events, something that is impossible in general with probabilities (the proposition "event A has twice the odds of event B" is always defined, whereas "event A is twice as probable as event B" would not be if $P(B) > 0.5$.) Likewise, the log-odds (also called a *logit*) span the whole real line, which make them a suitable parameterization of probabilities for models that output real numbers; as the simplest example, logistic regression represents a logit as a linear combination of input variables, $\text{logit(p)} = \boldsymbol{\theta}'\mathbf{x} + \epsilon$, where each variable contributes linearly to the log-odds (thence multiplicatively to the odds).



▶ **Figure 5.24.** *Schematic diagram of a Financial Neural Network. We assume an adjustable network weight on each arrow on the figure (except on the output). The free-standing arrows represent biases.*



All model parameters, namely $\boldsymbol{\theta}$ and the ζ_k, can be learned by maximum likelihood using the method of iteratively reweighted least squared (McCullagh and Nelder 1989).

5.5.2 Architecture

The architecture of a Financial Neural Network is depicted in Fig. 5.24. It can be seen both as a straightforward generalization of the previous proportional-odds model and as a multilayer perceptron with an output layer made up of a number of classifiers with an appropriate parameterization, and an input layer identical to that described in §3.1.2/p. 73. In the application that we shall consider, we obtain a three-way classification for each asset in the portfolio (long/neutral/short). Note that this model is more properly seen as a *trading model* rather than an *allocation model*, since it does not output a vector of portfolio weights but immediate trading decisions.

Let $\eta(\mathbf{x})$ be the linear combination of the hidden units (or the network inputs, if there are no hidden units, in which case we fall back on the proportional-odds model), represented in Fig. 5.24 by the "sigma node" in the *output layer* portion. As will become clear below, we can interpret this variable as a "latent asset pseudo-return", i.e. an indication of how well the asset will perform over the next investment horizon, according to the network. The purpose of the output layer is to partition the set of values that $\eta(\mathbf{x})$ can take (the real line) into ordered regions, each associated with one of the output classes. This is equivalent to defining cutoff points as in

eq. (5.21).

As for the proportional-odds model, the FNNET* learns a representation of the cumulative distribution of the class label as

*Financial Neural Network.

$$P(Y \leq k|\mathbf{x}) = \text{sigm}(\zeta_k - \eta(\mathbf{x})).$$

Obviously, the parameters ζ_k must be learned, and it is crucial to preserve the ordering relationship (5.21) for the proper definition of the model. The following parameterization was found to be very effective for transforming the set of ordering constraints into an unconstrained problem suitable for training with conjugate gradients optimization:

1. The first parameter, ζ_0 is unconstrained;
2. Each remaining parameter $\tilde{\zeta}_k$, also unconstrained, is incorporated as a *positive increment* to the previous threshold via a softplus transformation
$$\zeta_k = \zeta_{k-1} + \text{softplus}(\tilde{\zeta}_k), \qquad k = 1, \ldots, K,$$
where the softplus is defined by eq. (5.5) (p. 128).

This differs from previously-proposed parameterizations (e.g. Mathieson 1997) and results in easier model training (empirically, it is observed to be less sensitive to local minima).

The final position taken by the model is the combination of the nominal position associated with each class (i.e. $-1, 0, +1$, respectively for short, neutral and long positions), weighted by the posterior probability of the class.

Note that this formulation only considers a single asset at a time; in particular, whole-portfolio constraints are not considered. Risk constraints can be incorporated ex-post (e.g. by ensuring that the value-at-risk exposure never exceeds a given target). Total exposure constraints can also be incorporated in the training objectives described below.†

5.5.3 Network Training and Regularization Issues

Whereas traditional ordinal regression models are trained to maximize the in-sample likelihood, it is desirable to use a financial training criterion if our task is ultimately one of portfolio management. However, as will be clarified in §5.6.2/p. 171, a pure financial objective tends to exhibit relatively poor performance out of sample. Moreover, when transaction costs are taken into account, the issue of local minima in the optimization objective raises its head (§5.1.4/p. 130).

†All the experiments that we consider in this chapter are carried out on a single asset; for this reason, portfoliowise exposure constraints are not considered over those being applied for individual assets.

Financial Objective

In the experiments that we consider in this chapter, the financial criterion being optimized is the absolute Sharpe ratio ("absolute" in that it is expressed in terms of absolute portfolio returns, not an excess return over the risk-free rate), although others such as the Sortino ratio are just as easy to consider. Let $\mathbf{w}_{i,t-1}(\boldsymbol{\theta})$ be the position taken in asset i at the start of period t by the model*—where the dependence on adjustable model parameters $\boldsymbol{\theta}$ is made explicit—, and let $\mathbf{r}_{i,t}$ be the relative return of asset i during period t. Let $r_t = \sum_i \mathbf{w}_{i,t-1}(\boldsymbol{\theta})\mathbf{r}_{i,t}$ be the portfolio return for period t.

Mind the time indexing conventions of §1.2/p. 6.

The financial objective, which we *minimize* over the training set, is

$$J_{\text{FIN}}(\boldsymbol{\theta}) = -\frac{\bar{r}}{\sqrt{\hat{\sigma}_r^2 + \rho}} + \kappa J_{\text{TURNOVER}}(\boldsymbol{\theta}) \tag{5.23}$$

where $\bar{r}$ is the sample mean of portfolio returns over training time steps, $\hat{\sigma}_r^2$ is the sample variance of portfolio returns, ρ is a small positive value to avoid an indefinite 0/0 situation when the model stays neutral over the entire training period† and $J_{\text{TURNOVER}}(\boldsymbol{\theta})$ is a turnover penalty to discourage excessive trading

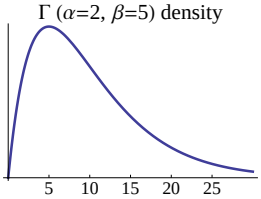
$$J_{\text{TURNOVER}}(\boldsymbol{\theta}) = \sum_t \sum_i |\mathbf{w}_{i,t}(\boldsymbol{\theta}) - \mathbf{w}_{i,t-1}(\boldsymbol{\theta})|. \tag{5.24}$$

In eq. (5.23), κ is a non-negative hyperparameter weighting the importance of turnover penalization.

Regularization

Several additional terms shape the objective function towards *a priori* desirable behavior.

▶ **Ensuring Good Threshold Dispersion :** The thresholds in the output layer are constrained to be increasing by the parameterization outlined in §5.5.2/p. 164. However, nothing otherwise prevents them from being very close or widely dispersed, both constituting situations of which escaping out may be difficult during the optimization. To this end, we follow the idea of Mathieson (1995) and incorporate a "gamma prior" on the *difference between successive thresholds*. The gamma distribution with shape parameter α and scale parameter β has probability density



$$p_\Gamma(x; \alpha, \beta) = \frac{e^{-\frac{x}{\beta}} x^{\alpha-1} \beta^{-\alpha}}{\Gamma(\alpha)},$$

where $\Gamma(\cdot)$ is Euler's gamma function. The "prior" is incorporated as the

†As might happen after initialization, if parameters are initialized to zero instead of random values. In experiments, the value $\rho = 10^{-4}$ was used.

log-gamma density (omitting constant terms) and applies to all threshold differences.* It takes the form

$$J_{\text{THRESH-PRIOR}}(\boldsymbol{\theta}) = \sum_{k=2}^{K} \Big(- \frac{\zeta_k - \zeta_{k-1}}{\beta} + (\alpha - 1) \log(\zeta_k - \zeta_{k-1}) \Big), \qquad (5.25)$$

where the thresholds ζ_k are part of the model parameter vector $\boldsymbol{\theta}$.†

Empirical explorations (not reported here) indicated that this term significantly helps find thresholds with adequate dispersion; when not present, it could occur that thresholds would remain "clumped" after training, appearing to be stuck in a poor local optimum.

▶ **Incorporating Explicit Targets at each Timestep :** The financial objective, by nature, imposes no constraint for what should be the position taken by the model at each training timestep. In particular, it offers no obvious mechanism by which the model should assimilate the target portfolio positions found by the K-best paths procedure. Although crude in appearance, we found effective to incorporate this information through a least-squares penalty term on the intermediate "latent" network output (the linear combination of the hidden units), denoted η in §5.5.2/p. 164. Let $\eta_{i,t}(\boldsymbol{\theta})$ the value of the intermediate network output for asset i at training timestep t given model parameters $\boldsymbol{\theta}$, and let $\mathbf{w}^*_{i,t}$ be the target weight for asset i at time t found by the K-best paths procedure. The penalty that we incorporate in the objective takes the form

$$J_{\text{TARGETS}}(\boldsymbol{\theta}) = \sum_{t} \sum_{i} \big(\mathbf{w}^*_{i,t} - \eta_{i,t}(\boldsymbol{\theta}) \big)^2. \qquad (5.26)$$

This has the effect of pulling the latent output towards the K-best target position (which, for the three-way classification, can be $-1, 0$ or $+1$). Other intermediate targets, such as asset returns over fixed horizons (e.g. one day or one month), produced little, if slightly worse, difference.‡

*Note that we speak of a "prior" with quotation marks by analogy with the form that this term would take under a maximum *a posteriori* training criterion (i.e. if we were training the model according to a regularized maximum likelihood objective). Since the financial objective presents no such direct interpretation, we slightly abuse language by speaking of a prior here, although its contribution to the objective function takes the same form.

†In our experiments, constant values $\alpha = 2$ and $\beta = 5$ were used, chosen after a cursory initial investigation but without exhaustive exploration of possible values.

‡Several alternative criteria were experimented with, but did not yield tangible improvement in ultimate model performance. In particular, inspired by the notion of *hints* (Abu-Mostafa 1995), we considered supplementary network outputs, only used during training, and that seek to predict next-period asset returns (for linear output units) or the action found by the K-best paths procedure (for multinomial output units). In both instances, appropriate terms, respectively a mean-squared error or negative log-likelihood, are added to the overall objective function to ensure that the additional network weights are trained appropriately. The motivation for these "artificial network limbs" is that some

Overall Objective

The overall FNNET optimization objective, which we seek to minimize, is the combination of the previously-outlined terms, *viz.*

$$\begin{aligned} J_{\text{FNNET}}(\boldsymbol{\theta}) = J_{\text{FIN}}(\boldsymbol{\theta}) \\ + \lambda_{\text{TP}} J_{\text{THRESH-PRIOR}}(\boldsymbol{\theta}) \\ + \lambda_{\text{TARG}} J_{\text{TARGETS}}(\boldsymbol{\theta}) \\ + \lambda_{\text{WD}} \|\boldsymbol{\theta}\|_2^2, \end{aligned} \tag{5.27}$$

where λ_{TP} and λ_{TARG} are hyperparameters controlling the impact of their respective terms in the objective. The last term, gated by the λ_{WD} hyperparameter, corresponds to a standard weight decay penalty (ℓ_2-norm of the parameter vector) frequently used in training neural networks (Bishop 1995).

One hypothesis—stated here but not otherwise fully tested in this work—with regards to the beneficial impact of the J_{TARGETS} term on out-of-sample performance, examined in §5.7/p. 173, concerns its influence on the Hessian of the objective function at the optimum: the squared-error terms in eq. (5.26) contribute a substantial and nonvanishing curvature to the overall objective function around a local minimum.* As a result, it has the probable impact of reducing variance in the parameter estimates, at the cost of some increased bias. More is said on the importance of regularization in §5.8/p. 189.

Network Prefitting

The unconstrained objective (5.27) can be optimized using gradient-based optimization from starting values of parameters $\boldsymbol{\theta}$. The algorithm of conjugate gradient descent (Bertsekas 2000) was found particularly effective for this task.† However, still due to the fundamental reasons uncovered in §5.1/p. 123, good starting parameters are crucial lest one be prepared to navigate the treacherous waters of local minima.

One approach that was found of considerable help (§5.7/p. 173) is to initialize parameters in a special "prefitting" stage that takes place before the full objective (5.27) is optimized. The criterion here is simply the cross-entropy loss (maximum likelihood under a multinomial distribution, see, e.g. Bishop 2006), a commonly-used criterion for classification problems.

Two major benefits arise from this procedure:

---

gradient information percolates from the secondary objective back to first-layer network weights, where they can influence the main training objective through the shared representation of the hidden units.

*"Substantial", given the value of λ_{TARG} that were found to yield good results.

†The specific forms of the financial objective (5.23) and turnover penalty (5.24) make it intrinsically a "batch" objective, unsuitable for the stochastic gradient methods that are generally found most effective for training neural networks (Orr and Müller 1998).

- Explicit targets can be used during prefitting; in particular the targets found by the K-best paths procedure can be put to use at this stage (which are compared to simpler targets arising from daily or monthly returns in §5.7/p. 173);

- The implicit network recurrence induced by the turnover penalty (5.24) is broken, allowing the network to train faster.

There is a compelling analogy between this approach and the flurry of recent results on the training of deep networks (Hinton, Osindero, and Teh 2006; Bengio, Lamblin, Popovici, and Larochelle 2007), where individual network components are trained independently according to (frequently unsupervised) relatively simple criteria; and only at a later stage, called "fine-tuning", the full objective is brought to the task to finalize network training.

5.6 Experimental Questions and Methodology

Given that we have established (§5.1/p. 123) that trajectories found by the K-best paths approach do indeed produce "better" targets (in the sense of maximizing a realized Sharpe ratio financial objective, accounting for transaction costs) than other optimization approaches, we wish to answer the following two experimental questions:

1. Can we make use of these targets within a learning algorithm, and obtain significantly better out-of-sample *financial performance* (as measured by the criterion of interest, e.g. Sharpe ratio) than models trained with simpler principles;

2. Does the ordinal regression framework for making portfolio decisions bring robustness, out of sample, compared to standard regression or classification settings.

5.6.1 Controller Architectures

In order to investigate these questions, we compare a number of model types, as described next. The remaining experimental setting (assets, etc.) is fully described in §5.6.2/p. 171. To set notation, let $\mathbf{x}_t$ be the set of input variables at time t and $p_t \in [-1, +1]$ a position to be taken in the asset (all experiments are performed on a "portfolio" with a single asset). This is a real number since we allow fractional positions to be taken. We use $\boldsymbol{\theta}$ to denote the model's adjustable parameters, if necessary. Finally, we use $\mathbf{X}$ and $\mathbf{y}$ to denote, respectively, the matrix of training inputs and vector of training targets provided to the model.

Target Types

The following target types are used with various models:

- `RETURN-DAILY`: daily asset return (regression target).
- `RETURN-MONTHLY`: monthly asset return (regression target).
- `SIGN-DAILY`: the sign of the daily asset return (classification target).
- `SIGN-MONTHLY`: the sign of the monthly asset return (classification target).
- `PURE-KBEST`: the position obtained after K-best rescoring, without any gradient-based fine-tuning (classification target).

Long-Only Model (LO)

This is the simplest model, which takes a constant long position:

$$p_t = +1.$$

It provides a standard benchmark against which to evaluate other models.

Linear Regression (LR)

This is a thresholded linear regression, where the position is obtained as follows:

$$y_t = \boldsymbol{\theta}' \mathbf{x}_t \tag{5.28}$$

$$p_t = \begin{cases} -1 & \text{if } \ y_t < -\gamma, \\ +1 & \text{if } \ y_t > +\gamma, \\ 0 & \text{otherwise,} \end{cases} \tag{5.29}$$

where γ is a threshold hyperparameter. Note that the linear model does not include an intercept; this was uniformly found to yield more robust out-of-sample performance. The possible training targets for this model are either `RETURN-DAILY` or `RETURN-MONTHLY`.

Gaussian Process (GP)

This is a thresholded regression Gaussian process (Rasmussen and Williams 2006),* where the position is obtained as

*Gaussian processes are fully described in Chapter 6.

$$y_t = K_{\boldsymbol{\theta}}(\mathbf{x}_t, X)(K_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I)^{-1}\mathbf{y} \tag{5.30}$$

$$p_t = \begin{cases} -1 & \text{if } \ y_t < -\gamma, \\ +1 & \text{if } \ y_t > +\gamma, \\ 0 & \text{otherwise,} \end{cases} \tag{5.31}$$

where γ is, as above, a threshold hyperparameter and σ_n^2 is a regularization hyperparameter (which can be interpreted as a noise level in sampling a latent function process). The function $K_{\boldsymbol{\theta}}(\cdot, \cdot)$ is a symmetric semi-positive definite *covariance function* (also known as a *kernel*) and is parameterized by the hyperparameters $\boldsymbol{\theta}$. The result when both arguments are a matrix is defined to be the matrix of pairwise kernel function evaluations between the rows of the matrices (the so-called Gram matrix); when one argument is a vector and the other a matrix, the result is defined to be the vector of pairwise evaluations between the vector and each row of the matrix.

The covariance function used in the present work is the *rational quadratic* function, which can be interpreted as an infinite mixture of Gaussian kernels, and defined as

$$K_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{x}') = \left(1 + \frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\alpha\sigma_\ell^2}\right)^{-\alpha},$$

where $\boldsymbol{\theta} \equiv (\alpha, \sigma_\ell^2)$ are hyperparameters, with α controlling the decay rate of the mixture components and σ_ℓ^2 is a length-scale parameter. The hyperparameters are learned by maximizing the marginal likelihood on the training data (see §6.2.3/p. 251).

Classification Neural Network (CNNet)

This is a standard multilayer neural network with one hidden layer, tanh activations on the hidden layer, and softmax activation on the output layer, trained to minimize the cross-entropy loss (Bishop 2006). It can be trained with either `SIGN-DAILY` or `SIGN-MONTHLY` targets.

Financial Neural Network (FNNet)

This is the model whose architecture is introduced in §5.5.2/p. 164. The training criterion is a regularized Sharpe Ratio, but as explained previously, additional regularization making use of `PURE-KBEST`, `SIGN-MONTHLY` or `SIGN-DAILY` targets can be provided (used both at the prefitting stage, and within the J_{TARGETS} term of the financial objective 5.27).

5.6.2 Experimental Plan

Financial markets are notoriously noisy, and it is obvious that a comparison between the above models that would cover a single market and time period would be wholly insufficient. To properly assess the performance difference with any confidence, an evaluation covering a number markets and spanning sufficient time is required. However, testing across several markets implies some caveats: foremost is that the definition of economic variables that are known to be predictive of market returns may vary from country to country (depending on how governmental agencies choose to report them), and some important variables may not be available at all. Second, in the less highly-developed markets, some variables may have been available only recently

Table 5.1. *Summary statistics for the three markets covered in the experiments.*

Market	Time Periods			Nb. Input Variables	
	First Obs.	Test Start	Last Obs.	Fundamental	All
U.S.	1983/01/03	1990/09/03	2007/12/31	15	20
Canada	1980/01/03	1991/01/02	2007/12/31	22	27
Europe	1980/01/03	1991/01/02	2007/12/31	19	24

and a level of ingenuity is required to "backward-extend" them (a procedure known as *backcasting*, as opposed to *forecasting*) to obtain series lengths meeting our minimum duration desideratum.

To this end, we focused on three major equity markets in the present experiments: the United States, Canada and Europe. Our financial instruments are, for the U.S., the S&P 500 futures (rolled every quarter on the March–June–September–December schedule, also known as the H M U Z schedule), for Canada the S&P/TSX Composite price index, and for Europe the Dow Jones Eurostoxx-50 price index.* Complete details regarding the methodology used to construct the input variables and assess their theoretical forecasting power are given in §5.B/p. 197; a summary of the date coverage is shown in Table 5.1. The available date ranges are listed, along with the number of input variables used in the "fundamental" versus "fundamental+technical" modes.

We follow an evaluation procedure based on sequential validation:

- The initial training set ranges from the date of the "first observation" to the day before the "test start" (*cf.* Table 5.1).
- Testing starts on the "test start" day.
- The sequential validation proceeds in one-year increments: the initial models are trained and tested for one year. That test data is then added to the training set, the models are retrained and tested for one more year and so forth, until the day of the "last observation".

Hyperparameter Selection

We used two complementary procedures to select hyperparameters and analyze results. The first one is the simplest: for each of the models covered in §5.6.1/p. 169 (in addition to FinancialNNet), we select the best set of hyperparameters, on the Sharpe ratio criterion, on the period ending in

*For the latter two, financial futures were not available extending sufficiently far in the past to use them as instruments; we assume that investing in the price index would have been possible over the entire period. This downward-biases our financial performance results slightly, since an admissible instrument, such as an Exchange-Traded Fund, would offer dividend payments in addition to price appreciation.

1998 (inclusively). The final test of model performance is given on the period 1999–2007. This approach is used for the country-specific results of §5.7.2/p. 177.

A different, somewhat more robust, technique is used for across-country results (§5.7.4/p. 186). For each model type and each market, we consider the set of hyperparameters giving "statistically equivalent" performance (using an analysis of variance, described in §5.A/p. 192) on the Sharpe ratio criterion (after transaction costs). More specifically, we identify the hyperparameter combination yielding the best overall performance ending in 1998 (inclusively), and enumerate all other combinations that would yield statistically *not significantly different* performance at the 95% level. This constitutes the "equivalence set". Then, for analysis purposes, we regard as "truly out-of-sample" the performance obtained on the period 1999–2007, averaging out the Sharpe ratios of all individual portfolio obtained by an hyperparameter combination that is part of the equivalence set. The justification for this latter mode of analysis is to provide an "expected Sharpe ratio" obtained from selecting a model at random among the set of well-performing hyperparameters.

Simulation Details

The financial simulation is carried out to a high level of detail, using the simulation techniques described in Chapter 4. All trades are performed at the daily opening price. Mark-to-market and performance evaluation (return computation) are carried out with daily closing prices. Transaction costs are accounted for, and it is assumed that proportional costs of 5 basis points are incurred for each trade. These costs are realistic given the liquidity of the traded instruments in their respective markets.

5.7 Experimental Results

The experimental results are presented as follows: first we summarize the large number of detailed results presented in the appendices (§5.7.1/p. 173). We follow by a country-specific financial performance analysis (§5.7.2/p. 177) and a statistical analysis of the financial performance difference between models (§5.7.3/p. 178). We finish by an attempt at combining individual results to obtain a big-picture understanding (§5.7.4/p. 186).

5.7.1 Summary of Detailed Results

Detailed experimental results for each country, model and time period appear in §5.C/p. 212 to §5.E/p. 233. A few general conclusions are drawn from the analysis performed in the appendix:

Table 5.2. *Model choice stability: which models perform best during the validation and test periods.*

Market	Best Model (Validation)	Best Model (Test)
U.S.	FinancialNNet	Linear Regressor
Canada	FinancialNNet, Linear Regressor	Classification NNet, Linear Regressor
Europe	FinancialNNet, Gaussian Process, Long-Only	FinancialNNet

Model and Hyperparameter Instability

Model instability is notorious in all of financial and economic modeling (see p. 90 for a brief review of the relevant literature), and is due both to the high noise levels in the data—which make overfitting a constant issue—and structural instability in the underlying generating processes. In our context, beyond stability at the parameter level (study of which we omit, since the extremely large number of models trained in the sequential validation framework make this analysis tedious), we have to contend with stability at the *hyperparameter* and *model choice* levels.

The latter is easier to summarize: Table 5.2 summarizes the best-performing models during the validation and test periods, across the studied markets. (By "best-performing" we refer to the overall analyses performed in §§5.C.5, 5.D.5 and 5.E.5, using the Sharpe ratio criterion.) Whereas the FinancialNNet is always among the best ones during the validation period, this significantly better performance is preserved during the test only for Europe. Likewise, the Linear Regressor keeps its standing only for Canada. This ranking difference may appear surprising since all individual models are well-regularized; moreover, the Linear Regressor, FinancialNNet and ClassificationNNet models have approximately the same capacity (the latter two are always selected to have zero hidden units), with a small number of parameters compared to the number of training examples (from 15 to less than 30 parameters against several thousands training examples). Their differences lie in the output-layer architecture and associated training criteria. As such, one is pressed to invoke nonstationarities in the underlying markets to explain the substantial model ranking variability.

The stability of individual model hyperparameters is summarized in Table 5.3. The following definition of stability is used: if a choice of hyperparameter is made during the first period, and would have been significantly wrong during the second (according to the analyses of variance shown in the appendix), this is deemed "unstable". The rest is considered "stable". Note that the case where we select a value as highly significant during the first period, only to have it not significantly different from others during the second period is deemed "stable" under this definition, according to the principle that the absence of evidence (of stability) should not be construed as evidence of absence.

At this level, the picture is more mixed and more model-dependent, even for the same hyperparameter across different models. We note that the

Table 5.3. *Stability of significant hyperparameters for each model, as defined by which hyperparameter value would be selected on each subperiod. See Table 5.29 (p. 212) for the definition of each hyperparameter. 'S'=Stable; 'U'=Unstable; '–' means that a hyperparameter was not evaluated for a given country.*

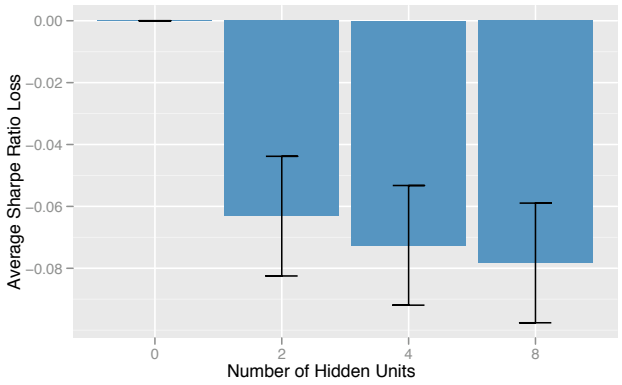
Model	Hyperparameter	Country			Summary
		US	CA	EU	
Linear Regressor	`EquityIndexes.input.flavor`	S	U	U	Mostly Unstable
	`EquityIndexes.rectify.thresh`	U	S	U	Mostly Unstable
	`EquityIndexes.target.init`	–	S	S	Stable
Classif. NNet	`EquityIndexes.input.flavor`	–	S	U	(Mixed)
	`EquityIndexes.target.init`	S	U	U	Mostly Unstable
	`Classification.nb.classes`	S	S	S	Stable
	`OptClassifNN.nhidden`	S	S	S	Stable
	`OptClassifNN.weight.decay`	S	U	U	Mostly Unstable
Gaussian Process	`EquityIndexes.input.flavor`	S	S	U	Mostly Stable
	`EquityIndexes.target.init`	S	S	S	Stable
	`kernel.options.hyper.ard`	U	–	–	(Unstable)
	`optimizer.options.nstages`	S	S	S	Stable
Financial NNet	`EquityIndexes.input.flavor`	S	U	S	Mostly Stable
	`EquityIndexes.target.init`	U	S	S	Mostly Stable
	`KBest.K`	S	–	–	Stable
	`OpTrProblem.prop.cost`	S	S	–	Stable
	`OptFinancialNNet.hint.penalty`	U	S	S	Mostly Stable
	`OptFinancialNNet.prefit.nstages`	–	S	S	Stable
	`OptFinancialNNet.prop.cost`	S	U	S	Mostly Stable
	`OptFinancialNNet.weight.decay`	–	S	S	Stable

Linear Regressor and ClassificationNNet tend to be unstable, whereas the opposite holds for the Gaussian Process and FinancialNNet.

Bad Performance of Higher-Capacity Models

Standard machine learning models (Gaussian processes, classification neural networks with two or more hidden units) consistently exhibit relatively poor performance. This suggests strongly that for this task, higher-capacity models have worse performance. This effect is illustrated for the Classification Neural Network in Fig. 5.25, which shows the average *Sharpe ratio loss* of using increased number of hidden units, compared to a baseline of zero hidden units (for which the loss is zero). This average is taken across all markets and time periods. As seen, with as little as two hidden units, a Sharpe ratio loss of 6% is incurred, an extremely statistically significant degradation. In addition, greater numbers of hidden units monotonically decrease performance.

Closely similar results—monotonic performance degradation with an in-



◀ **Figure 5.25.** *Average Sharpe ratio loss of using additional hidden units with the Classification Neural Network, with respect to a baseline of zero, across all markets and the 1990–2007. Error bars represent 95% confidence intervals around the mean loss.*

creasing number of hidden units—were also measured for the Financial Neural Network (not illustrated here).

Importance of Network Prefitting

With FinancialNNet, we note that network prefitting is important—i.e. initial training with a classification criterion to adequately initialize the network weights. In fact, performance without prefitting is terrible, suggesting that local minima in the overall financial objective function are prevalent and difficult to overcome. This can be explained by noting that the prefitting criterion provides explicit model targets—and hence a lot of information—at each training timestep, whereas the financial criterion consists of a single objective for the entire training period, making credit assignment to individual decisions more difficult.

On the other hand, the financial criterion is also important, as performance using *only* prefitting is also bad.

Impact of Various Terms in FinancialNNet Objective

Regarding the impact of the various parameters within the FinancialNNet objective (*cf.* eq. (5.27)), we note the following:

- *Threshold-dispersion prior* (λ_{TP}): we did not extensively experiment with this parameter, but some preliminary experiments showed that it helps avoid bad solutions.

- *Target hint* (λ_{TARG}): a positive value always yields a better performance.* The precise optimum value varies slightly for each market, but is in the range of 10 BP ±5 BP. The target type used within the hint term (i.e. whether `PURE-KBEST`, `SIGN-MONTHLY` or `SIGN-DAILY`) is not significant for Canada and Europe, and unstable for the United States

*Results using a zero value are not reported, since they consistently led to much worse outcomes in all experiments.

(insignificant during the validation period, but favoring `SIGN-DAILY` during the test).

- *Weight decay* (λ_{WD}): this parameter shows a clear and consistent optimum (10^{-4} to 10^{-6}).

Transaction Costs and FinancialNNet Performance

Transaction costs are used at two different levels within FinancialNNet and have a substantial impact on performance. The first such hyperparameter, `OpTrProblem.prop.cost`, plays its role within the K-best search, where a value in the range of 1%–1.5% works best. The second hyperparameter is used to penalize turnover within FinancialNNet optimization (it is not on a true scale of transaction costs); here a value of 10 BP–20 BP yields the best results.

Beyond specific parameter values, a conclusion to draw is that incorporating transaction costs within the training objective consistently improves test performance.

On Regularization

A high-level message is that **model regularization is crucial to good performance**. But regularization takes a variety of forms (e.g. all terms in addition to the main financial objective in eq. (5.27)), and should not only be interpreted restrictively in terms of a simple weight decay penalty. The good performance of linear regression can also be explained by its highly regularized nature. Connections can also be made with the imposition of constraints in classical mean–variance allocation, as discussed in §2.1.8/p. 31.

5.7.2 Country-Specific Financial Performance

Country-specific financial performance appear in Tables 5.5 to 5.10 and Figures 5.26 to 5.31. As mentioned above, the period ending in 1998 (inclusively) is used to select the best-performing hyperparameters (on the Sharpe ratio criterion), with the last period (1999–2007) providing an out-of-sample test. An explanation of the financial performance measures is provided in Table 5.4.

It is significant to note that the market conditions shifted dramatically between the validation and test periods: the 1990's represented one of the greatest bull markets of the twentieth century, and the early 2000's a quite dire bear market. As such, a model that exhibits good generalization performance in this context is performing a non-trivial task (especially if it is able to correctly short the 2001–2002 bear market).

The overall picture is mixed, with no single model dominating all markets and time periods. For the United States (Tables 5.5 and 5.6 and Figs. 5.26

and 5.27),* the Linear Regression, Classification Neural Network and Financial Neural Network all perform quite well, with the first one performing best out of sample. All three models outperform the buy-and-hold (long-only) over both the validation and test periods.

For Canada (Tables 5.7 and 5.8 and Figs. 5.28 and 5.29), all active models perform comparably, and outperform buy-and-hold. In particular, the Linear Regression and FinancialNNet models have been successfully capable of shorting the bear market of 2001–2002. The good performance of some models (e.g. Linear Regression and Gaussian Process) is attributable, in part, to exceptional overperformance in 2000: at that time, Nortel Network's capitalization came to represent about one third of the overall Canadian market, and the precipitous decline in Nortel's stock value starting in September of that year let to a consequent significant decline in the index. A well-positioned short, in this context, would have been the best course of action, and was that followed—to a first approximation—by the Linear Regression model.

Finally, for Europe (Tables 5.9 and 5.10 and Figs. 5.30 and 5.31), the most striking result is the manifest underperformance of the Linear Regression Model compared to the FinancialNNet, both of which have similar capacity.

We note that for all three markets, the FinancialNNet consistently appears close to the top; this is explored in the following pages.

5.7.3 Sharpe Ratio Difference Significance Tests

Of the results presented in the previous tables, one may wonder which of the differences are statistically significant. We applied the robust bootstrap test of Ledoit and Wolf (2008) to evaluate the significance of the measured Sharpe Ratio differences between all pairs of models. The test is summarized in §5.A.1/p. 192.

Tables 5.11–5.13 give the average Sharpe Ratio difference, computed on *daily returns*,† along with the p-value of this difference. Five thousands bootstrap repetitions were performed in all cases. Running the test on monthly returns instead of daily ones yields very similar results, including in the obtained p-values. (These p-values are not adjusted for multiple comparisons, an issue explained in §5.A.3/p. 196; the usual caveats apply.)

As is common in formal statistical comparisons of financial performance, the large variance in realized returns across strategies makes it difficult to achieve statistical significance in most measured differences. In the tables,

*Sharp-eyed readers may observe that the long-only cumulative return chart in Fig. 5.26 appears somewhat different from the S&P 500 chart that one is used to seeing; this is because the traded instruments for the United States are S&P 500 futures contracts rolled over one month prior to expiration, and not directly the price index. Were the latter used, the long-only chart would, obviously, become identical to the common expectation.

†A quick-and-dirty way to annualize the differences is to multiply them by $\sqrt{252}$, where 252 is the (average) number of trading days in a year.

Table 5.4. *Summary of the financial performance measures reported in Tables 5.5, 5.7, and 5.9. In the computations below, we assume that the strategy to be evaluated yields a sequence of monthly relative returns r_t, $t = 1, \ldots, T$. We define the cumulative return to time t as $C_t = \prod_{\tau=1}^{t}(1 + r_\tau)$.*

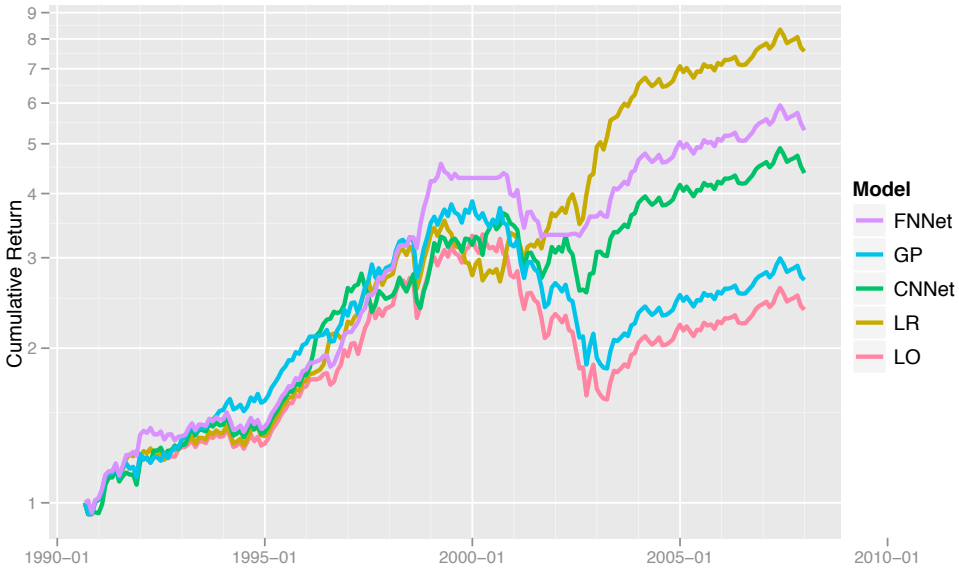
Name	Description	Expression
Annualized Return	Compound average yearly growth rate	$\text{CAGR} = C_T^{12/T}$
Avg Annual Return	Arithmetic average of returns (annualized)	$\hat{\mu}_a = 12\hat{\mu}_m$
Avg Annual Stddev	Standard deviation of returns (annualized)	$\hat{\sigma}_a = \sqrt{12}\hat{\sigma}_m$
Annual Sharpe Ratio	Absolute Sharpe Ratio (annualized)	$\text{SR} = \hat{\mu}_a/\hat{\sigma}_a$
Avg Monthly Return	Arithmetic average of returns	$\hat{\mu}_m = \frac{1}{T}\sum_{t=1}^{T} r_t$
Avg Monthly Stddev	Standard deviation of monthly returns	$\hat{\sigma}_m = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(r_t - \hat{\mu}_m)^2}$
Skewness	Sample skewness of monthly returns	$S = \frac{1}{T\hat{\sigma}_m^3}\sum_{t=1}^{T}(r_t - \hat{\mu}_m)^3$
Excess Kurtosis	Excess kurtosis of monthly returns	$K = \frac{1}{T\hat{\sigma}_m^4}\sum_{t=1}^{T}(r_t - \hat{\mu}_m)^4 - 3$
Best Month	Best monthly return	$\text{BM} = \max r_t$
Worst Month	Worst monthly return	$\text{WM} = \min r_t$
Fraction Months Up	Proportion of months with a positive return	$\text{FU} = \frac{1}{T}\sum_{t=1}^{T} I[r_t \geq 0]$
Maximum Drawdown	Worst peak-to-through return	$\text{DD} = \min_{t_1, t_2 > t_1} C_{t_2}/C_{t_1} - 1$
Drawdown Duration	Number of days from the peak (time $t1$ in previous expression) and the next day until which we reach the cumulative return C_{t_1}	
Drawdown From	Date corresponding to t_1	
Drawdown Until	Date at which we reach back the level last reached at C_{t_1}	

Table 5.5. *Financial Performance for the United States. The hyperparameters for each model are selected using financial performance on the 1990–1998 validation period. LO=Long-Only model, LR=Linear Regression, CNNet=Classification Neural Network, GP=Gaussian Process, FNNet=Financial Neural Network. Dates are given in the yy-mm-dd order.*

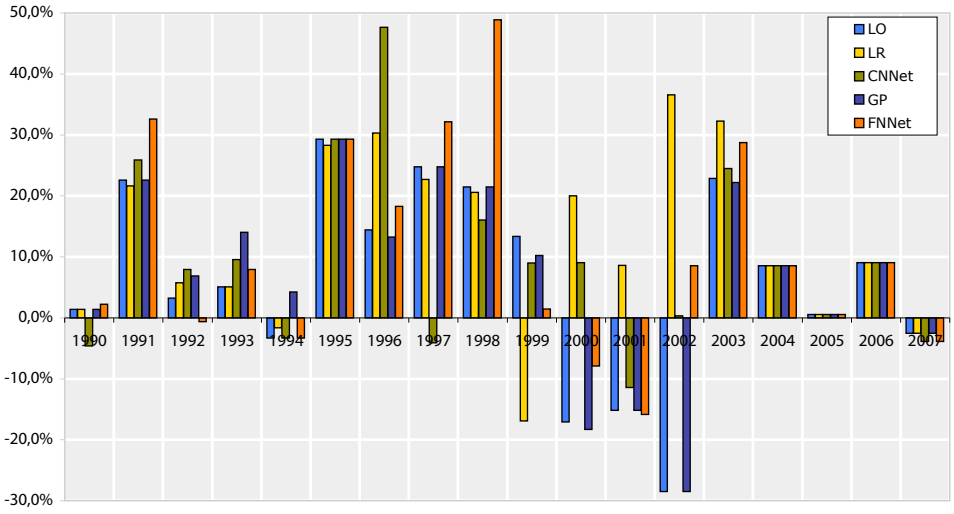
	1990–1998					**1999–2007**				
	LO	LR	CNNet	GP	FNNet	LO	LR	CNNet	GP	FNNet
Annualized Return	13.7%	15.6%	13.7%	16.3%	18.9%	−2.3%	9.5%	4.6%	−2.8%	2.6%
Avg Annual Return	13.7%	15.2%	13.7%	15.9%	18.0%	−1.3%	10.0%	5.4%	−1.9%	3.0%
Avg Annual Stddev	12.7%	11.8%	12.6%	12.6%	10.8%	14.1%	13.3%	13.3%	13.9%	9.5%
Annual Sharpe Ratio	1.08	1.29	1.09	1.26	1.67	−0.09	0.75	0.41	−0.13	0.31
Avg Monthly Return	1.1%	1.3%	1.1%	1.3%	1.5%	−0.1%	0.8%	0.5%	−0.2%	0.2%
Avg Monthly Stddev	3.7%	3.4%	3.6%	3.6%	3.1%	4.0%	3.8%	3.8%	4.0%	2.7%
Skewness	−0.762	−1.143	−0.268	−0.889	0.142	−0.356	0.235	0.147	−0.401	−0.429
Excess Kurtosis	2.700	3.995	1.005	3.023	0.553	0.330	0.700	0.378	0.315	2.064
Best Month	10.8%	8.1%	10.8%	10.8%	11.0%	8.8%	13.0%	10.0%	8.6%	8.1%
Worst Month	−15.0%	−15.0%	−11.6%	−15.0%	−5.4%	−11.9%	−8.9%	−9.9%	−11.9%	−9.5%
Fraction Months Up	69.0%	69.0%	64.0%	72.0%	71.0%	52.8%	62.0%	55.6%	51.9%	48.1%
Maximum Drawdown	−20.2%	−20.6%	−20.3%	−20.2%	−12.4%	−56.2%	−27.6%	−39.7%	−56.3%	−36.8%
Drawdown Duration	159	160	146	159	51	1082	931	1135	1334	1728
Drawdown From	98-07-17	98-07-16	98-07-30	98-07-17	98-08-25	00-03-24	99-04-27	00-11-08	99-07-16	99-03-31
Drawdown Until	98-12-23	98-12-23	98-12-23	98-12-23	98-11-15	—	01-11-13	03-12-18	—	03-12-23

Table 5.6. *Annual Returns for the United States.*

	LO	LR	CNNet	GP	FNNet		LO	LR	CNNet	GP	FNNet
1990	1.4%	1.4%	−4.6%	1.4%	2.3%	1999	13.4%	−16.9%	9.0%	10.2%	1.5%
1991	22.6%	21.6%	25.9%	22.6%	32.6%	2000	−17.1%	20.0%	9.0%	−18.3%	−7.9%
1992	3.2%	5.7%	7.9%	6.9%	−0.6%	2001	−15.2%	8.6%	−11.4%	−15.2%	−15.9%
1993	5.1%	5.1%	9.5%	14.0%	7.9%	2002	−28.5%	36.6%	0.3%	−28.5%	8.5%
1994	−3.3%	−1.6%	−3.3%	4.2%	−3.3%	2003	22.9%	32.3%	24.5%	22.2%	28.7%
1995	29.3%	28.3%	29.3%	29.3%	29.3%	2004	8.5%	8.5%	8.5%	8.5%	8.5%
1996	14.4%	30.3%	47.7%	13.3%	18.3%	2005	0.6%	0.6%	0.6%	0.6%	0.6%
1997	24.8%	22.7%	−4.1%	24.8%	32.2%	2006	9.0%	9.0%	9.0%	9.0%	9.0%
1998	21.5%	20.6%	16.0%	21.5%	48.9%	2007	−2.5%	−2.5%	−3.9%	−2.5%	−3.9%



◀ **Figure 5.26.** *United States: Cumulative return of the selected model of each type (log-scale). The 1990–1998 period is used for model selection and 1999–2007 is the final test.*



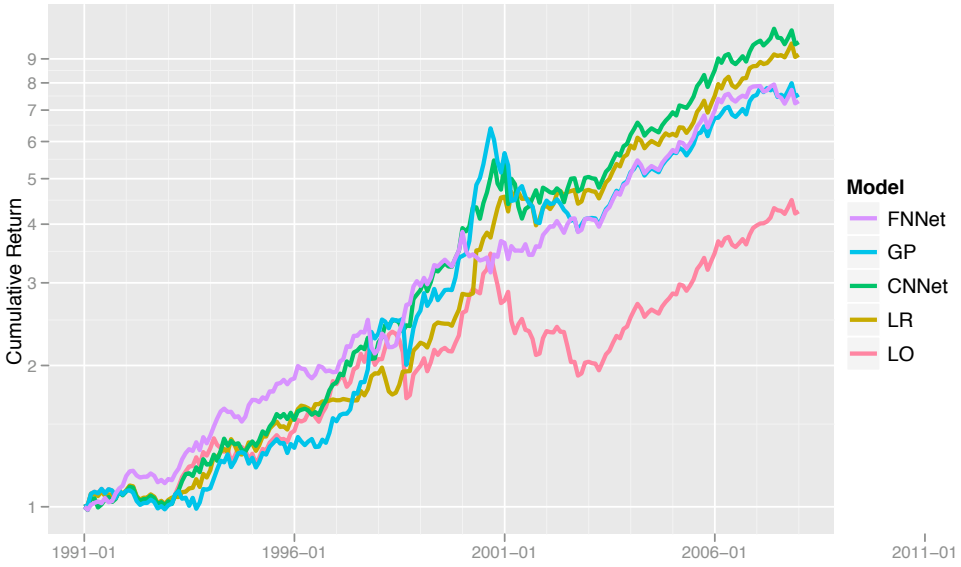
◀ **Figure 5.27.** *United States: Annual returns of the selected model of each type. The 1990–1998 period is used for model selection and 1999–2007 is the final test.*

Table 5.7. *Financial Performance for Canada. The hyperparameters for each model are selected using financial performance on the 1991–1998 validation period. LO=Long-Only model, LR=Linear Regression, CNNet=Classification Neural Network, GP=Gaussian Process, FNNet=Financial Neural Network.*

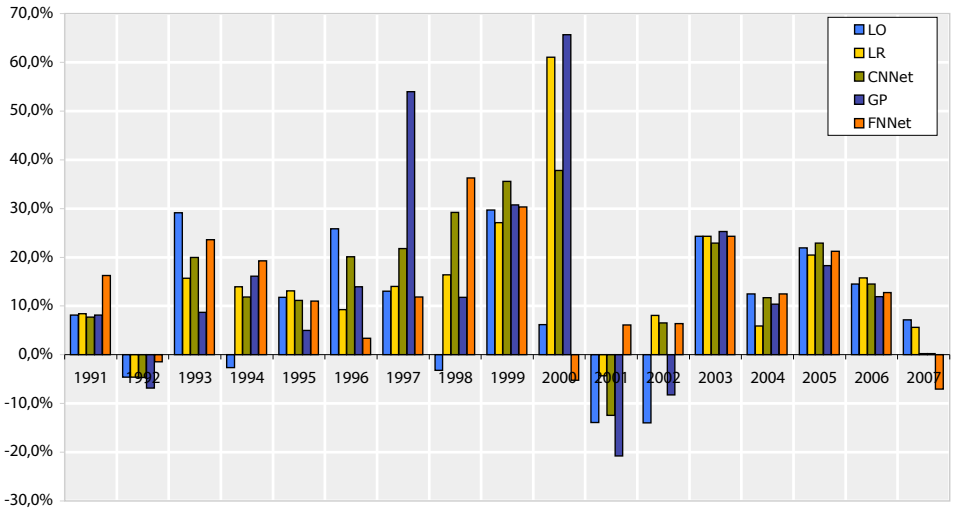
	1991–1998					**1999–2007**				
	LO	LR	CNNet	GP	FNNet	LO	LR	CNNet	GP	FNNet
Annualized Return	9.0%	10.6%	14.2%	12.8%	14.5%	8.8%	17.0%	14.5%	12.5%	10.6%
Avg Annual Return	9.7%	10.6%	14.4%	13.3%	14.6%	9.5%	16.7%	15.0%	13.1%	10.8%
Avg Annual Stddev	14.1%	10.0%	11.8%	15.6%	11.3%	14.5%	13.7%	16.5%	16.1%	12.1%
Annual Sharpe Ratio	0.69	1.06	1.22	0.85	1.29	0.66	1.22	0.91	0.82	0.89
Avg Monthly Return	0.8%	0.9%	1.2%	1.1%	1.2%	0.8%	1.4%	1.2%	1.1%	0.9%
Avg Monthly Stddev	4.0%	2.9%	3.4%	4.5%	3.3%	4.2%	3.9%	4.7%	4.6%	3.5%
Skewness	−1.324	0.470	0.303	−0.219	−0.408	−0.532	1.373	−0.573	−0.044	−0.292
Excess Kurtosis	5.982	0.747	1.677	6.635	2.295	0.754	7.709	2.392	2.054	0.228
Best Month	10.6%	10.6%	14.2%	20.1%	8.8%	11.8%	23.4%	14.1%	17.3%	9.6%
Worst Month	−20.2%	−6.3%	−9.8%	−20.2%	−12.7%	−13.3%	−8.5%	−18.4%	−16.0%	−8.9%
Fraction Months Up	61.1%	57.9%	63.2%	61.1%	64.2%	61.1%	63.9%	62.0%	64.8%	61.1%
Maximum Drawdown	−31.8%	−15.4%	−24.1%	−22.9%	−28.8%	−50.0%	−20.6%	−30.8%	−45.1%	−18.6%
Drawdown Duration	166	6	119	122	301	1950	624	1097	1909	707
Drawdown From	98-04-22	97-11-14	97-10-22	98-07-24	97-10-07	00-09-01	01-04-18	00-10-12	00-09-14	99-12-29
Drawdown Until	—	97-11-20	98-02-18	98-11-23	98-08-04	06-01-03	03-01-02	03-10-14	05-12-06	01-12-05

Table 5.8. *Annual Returns for Canada.*

	LO	LR	CNNet	GP	FNNet		LO	LR	CNNet	GP	FNNet
						1999	29.7%	27.1%	35.6%	30.7%	30.3%
1991	8.2%	8.4%	7.7%	8.2%	16.2%	2000	6.2%	61.0%	37.8%	65.6%	−5.2%
1992	−4.6%	−4.6%	−4.6%	−6.8%	−1.4%	2001	−13.9%	−4.3%	−12.5%	−20.8%	6.1%
1993	29.2%	15.7%	20.0%	8.7%	23.6%	2002	−14.0%	8.1%	6.5%	−8.3%	6.4%
1994	−2.6%	14.0%	11.8%	16.1%	19.3%	2003	24.3%	24.3%	22.9%	25.3%	24.3%
1995	11.8%	13.1%	11.2%	5.0%	11.0%	2004	12.5%	5.9%	11.7%	10.4%	12.5%
1996	25.8%	9.3%	20.1%	13.9%	3.4%	2005	21.9%	20.4%	22.9%	18.3%	21.2%
1997	13.0%	14.0%	21.8%	53.9%	11.8%	2006	14.5%	15.8%	14.5%	11.9%	12.7%
1998	−3.2%	16.4%	29.2%	11.8%	36.3%	2007	7.2%	5.6%	0.2%	0.2%	−7.1%



◀ **Figure 5.28.** *Canada: Cumulative return of the selected model of each type (log-scale). The 1991–1998 period is used for model selection and 1999–2007 is the final test.*



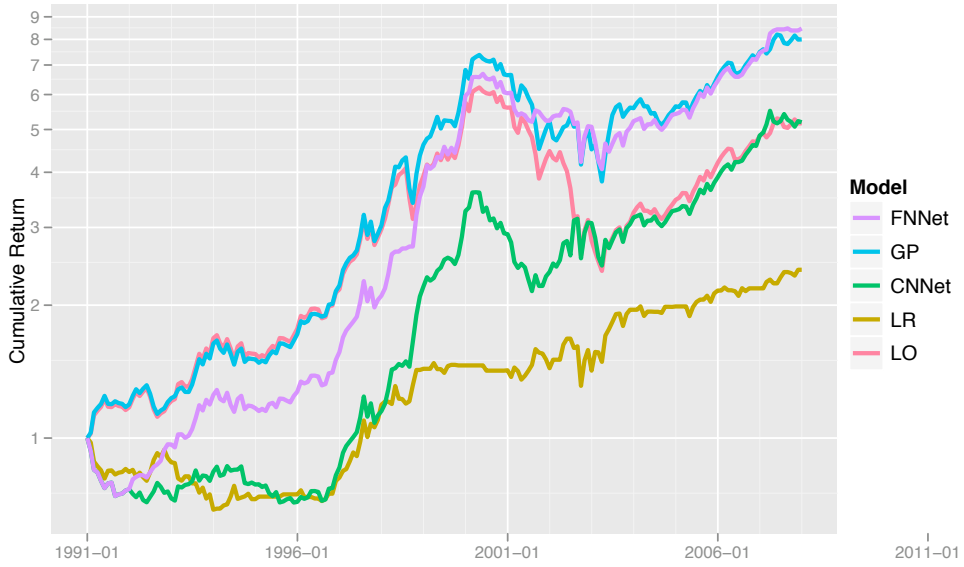
◀ **Figure 5.29.** *Canada: Annual returns of the selected model of each type. The 1991–1998 period is used for model selection and 1999–2007 is the final test.*

Table 5.9. *Financial Performance for Europe. The hyperparameters for each model are selected using financial performance on the 1991–1998 validation period. LO=Long-Only model, LR=Linear Regression, CNNet=Classification Neural Network, GP=Gaussian Process, FNNet=Financial Neural Network.*

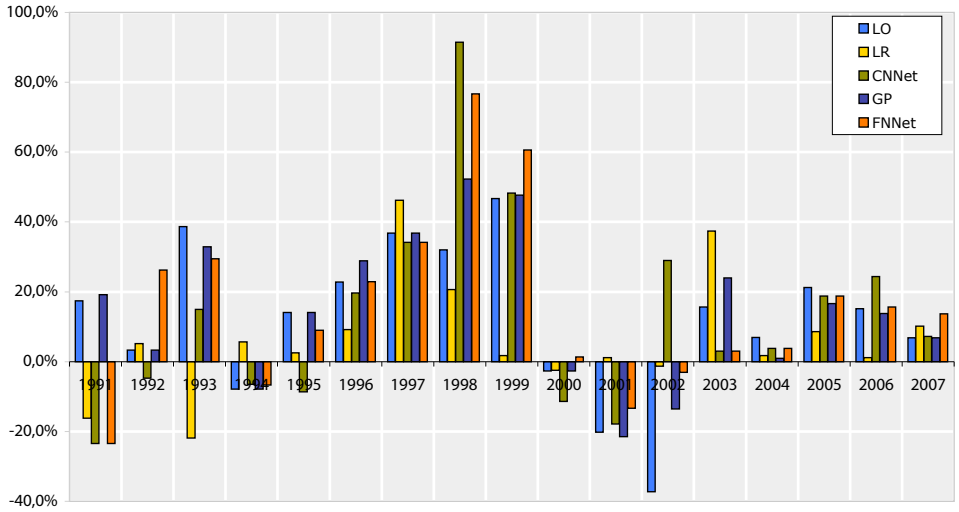
	1991–1998					**1999–2007**				
	LO	LR	CNNet	GP	FNNet	LO	LR	CNNet	GP	FNNet
Annualized Return	18.6%	4.6%	10.4%	21.1%	17.8%	3.1%	5.9%	10.0%	6.3%	9.6%
Avg Annual Return	18.4%	5.8%	12.1%	20.5%	18.7%	4.8%	6.7%	11.3%	7.8%	10.5%
Avg Annual Stddev	16.4%	13.5%	16.0%	17.0%	15.6%	18.4%	13.8%	18.6%	18.5%	16.2%
Annual Sharpe Ratio	1.12	0.43	0.76	1.21	1.20	0.26	0.49	0.61	0.42	0.65
Avg Monthly Return	1.5%	0.5%	1.0%	1.7%	1.6%	0.4%	0.6%	0.9%	0.7%	0.9%
Avg Monthly Stddev	4.7%	3.9%	4.6%	4.9%	4.5%	5.3%	4.0%	5.3%	5.3%	4.7%
Skewness	−0.590	0.206	0.364	−0.241	0.200	−0.360	−0.241	0.119	−0.247	−0.057
Excess Kurtosis	0.845	0.732	1.046	1.236	1.568	1.606	6.521	2.389	1.566	3.271
Best Month	11.5%	11.5%	14.9%	16.3%	17.9%	14.3%	14.7%	20.0%	15.2%	14.6%
Worst Month	−14.4%	−9.4%	−10.0%	−14.9%	−10.0%	−18.6%	−18.6%	−18.6%	−18.6%	−18.6%
Fraction Months Up	65.3%	47.4%	64.2%	64.2%	65.3%	56.5%	38.0%	62.0%	58.3%	57.4%
Maximum Drawdown	−34.1%	−34.4%	−32.2%	−24.6%	−28.9%	−66.2%	−25.1%	−49.2%	−50.2%	−46.7%
Drawdown Duration	80	2373	2310	116	784	1101	249	2038	2493	2215
Drawdown From	98-07-20	91-01-14	91-01-14	98-07-30	91-01-14	00-03-06	02-08-22	00-03-06	00-03-06	00-03-06
Drawdown Until	—	97-07-14	97-05-12	98-11-23	93-93-08	—	03-04-28	05-10-04	07-01-02	06-03-30

Table 5.10. *Annual Returns for Europe.*

	LO	LR	CNNet	GP	FNNet		LO	LR	CNNet	GP	FNNet
						1999	46.7%	1.8%	48.3%	47.7%	60.6%
1991	17.4%	−16.2%	−23.4%	19.2%	−23.4%	2000	−2.7%	−2.5%	−11.4%	−2.7%	1.3%
1992	3.4%	5.2%	−4.7%	3.4%	26.3%	2001	−20.2%	1.1%	−17.9%	−21.4%	−13.4%
1993	38.7%	−21.9%	15.0%	32.9%	29.5%	2002	−37.3%	−1.3%	29.0%	−13.5%	−3.1%
1994	−7.9%	5.6%	−6.5%	−7.9%	−6.7%	2003	15.7%	37.4%	3.0%	24.0%	3.0%
1995	14.1%	2.5%	−8.6%	14.1%	9.0%	2004	6.9%	1.7%	3.8%	1.0%	3.8%
1996	22.8%	9.2%	19.7%	28.8%	22.9%	2005	21.3%	8.6%	18.8%	16.6%	18.8%
1997	36.8%	46.2%	34.2%	36.8%	34.2%	2006	15.1%	1.2%	24.3%	13.8%	15.7%
1998	32.0%	20.6%	91.5%	52.3%	76.7%	2007	6.8%	10.2%	7.2%	6.8%	13.7%



◀ **Figure 5.30.** *Europe: Cumulative return of the selected model of each type (log-scale). The 1991–1998 period is used for model selection and 1999–2007 is the final test.*



◀ **Figure 5.31.** *Europe: Annual returns of the selected model of each type. The 1991–1998 period is used for model selection and 1999–2007 is the final test.*

we choose to use the 10% level as the threshold for significance, since no difference is significant at better than the 5% level. A "tournament" summary of those results appears in Table 5.14, which shows the number of times that each model is "beaten" by another over each time period, and the number of times this can be considered significant. We note that all active models perform better than the long-only model over both time periods. The model performing best during the test period is the Linear Regression, albeit it has one of the worst performances of the validation period. The best "overall" model (assuming it makes sense to compare the validation and test periods) is the FinancialNNet, and importantly, the latter never loses to another in a statistically significant way, which no other model achieves.

5.7.4 Pooling Performance Results

Rather than picking the single best value of hyperparameters over the validation period, we can rely on the "equivalence set" approach as described previously for hyperparameter choice. This is the methodology followed in the appendix for identifying good-performing subsets of hyperparameters. There are several alternatives for evaluating the resulting set of models, and a simple one is to average out the Sharpe ratios of all individual models.

In order to get a better picture of the performance variability across models and settings, we fit a linear mixed-effects model (McCulloch, Searle, and Neuhaus 2008; Pinheiro and Bates 2000), where the fixed effect is the model type and the random effect is a "default" performance level for a given country and time period. The overall model tries to explain the observed measured Sharpe ratio for the country-period pair i and model j as

$$S_{i,j} = \alpha + \beta_i + \gamma_j + \epsilon_{i,j}$$

where α is a global intercept, β_i is a random effect due to a given observed pair of country and time period (these effects are assumed to be suffered by all models), γ_j is a fixed effect due to model type and $\epsilon_{i,j}$ is a residual. In the model diagnostics presented in Table 5.15, the `Controller =` rows list the values of the fixed effects as an average Sharpe ratio in excess of the `ClassificationNNet` baseline model (whose performance is absorbed in the intercept, α).

We observe that the baseline performance exhibited by `ClassifNNet`, with a Sharpe ratio of 22%, is considered very statistically significant. Figure 5.32 shows that the residuals of the model are reasonably well approximated by the normal distribution (including a non-rejection of the null by a Jarque and Bera (1980) test), suggesting that the p-values obtained for the fixed effects shown previously should be reliable. From the results, we note that the FinancialNNet results exhibit substantially better performance than the "default level" implied by the random effect ($p = 0.074$), and such a level is not reached by any competing model. Although not extremely powerful at this stage, these results suggest that the FinancialNNet model

Table 5.11. *For United States, pairwise Sharpe ratio difference (computed on **daily returns**) between all pairs of models and p-value of this difference (in parentheses) under the robust test of Ledoit and Wolf (2008). The difference is row-model minus column-model; a positive number indicates that the row model is better than the column model. Entries significant at the 10% level are in bold. LO=Long-Only model, LR=Linear Regression, CNNet=Classification Neural Network, GP=Gaussian Process, FNNet=Financial Neural Network.*

	1990–1998					1999–2007			
	LR	CNNet	GP	FNNet		LR	CNNet	GP	FNNet
LO	−0.0076	0.0006	−0.0089	**−0.0278**	LO	**−0.0397**	−0.0231	0.0019	−0.0178
	(0.6513)	(0.9786)	(0.1686)	**(0.0566)**		**(0.0956)**	(0.2058)	(0.5496)	(0.2815)
LR		0.0082	−0.0013	−0.0202	LR		0.0166	**0.0415**	0.0218
		(0.6587)	(0.945)	(0.215)			(0.3343)	**(0.0812)**	(0.2601)
CNNet			−0.0095	−0.0284	CNNet			0.0249	0.0052
			(0.6227)	(0.1044)				(0.1668)	(0.7123)
GP				−0.0189	GP				−0.0197
				(0.2374)					(0.2328)

Table 5.12. *For Canada, pairwise Sharpe ratio difference (computed on **daily returns**) between all pairs of models and p-value of this difference (in parentheses).*

	1991–1998					1999–2007			
	LR	CNNet	GP	FNNet		LR	CNNet	GP	FNNet
LO	−0.012	−0.019	−0.0124	−0.0223	LO	−0.0372	−0.0189	−0.0127	−0.009
	(0.6141)	(0.3527)	(0.4567)	(0.3559)		(0.1052)	(0.4259)	(0.3923)	(0.7321)
LR		−0.007	−0.0004	−0.0103	LR		0.0183	0.0245	0.0282
		(0.7387)	(0.98)	(0.6457)			(0.4199)	(0.2831)	(0.155)
CNNet			0.0066	−0.0033	CNNet			0.0062	0.0099
			(0.7866)	(0.8448)				(0.787)	(0.5735)
GP				−0.0099	GP				0.0037
				(0.7237)					(0.8774)

Table 5.13. *For Europe, pairwise Sharpe ratio difference (computed on **daily returns**) between all pairs of models and p-value of this difference (in parentheses).*

	1991–1998					1999–2007			
	LR	CNNet	GP	FNNet		LR	CNNet	GP	FNNet
LO	0.0431	0.0263	−0.008	−0.0094	LO	−0.0114	−0.0174	−0.0082	−0.0192
	(0.2416)	(0.3255)	(0.4625)	(0.6503)		(0.6015)	(0.3325)	(0.3951)	(0.133)
LR		−0.0168	−0.0511	**−0.0525**	LR		−0.006	0.0031	−0.0078
		(0.4939)	(0.134)	**(0.0638)**			(0.7562)	(0.8794)	(0.6745)
CNNet			−0.0343	**−0.0357**	CNNet			0.0091	−0.0018
			(0.186)	**(0.0674)**				(0.6315)	(0.9062)
GP				−0.0014	GP				−0.0109
				(0.9428)					(0.4535)

Table 5.14. *Number of times that each model is "beaten" by another in the Sharpe ratio difference tables 5.11–5.13. The number of statistically significant beatings is in parentheses.*

	1990–1998		1999-2007		Total	
LO	9	(1)	11	(1)	20	(2)
LR	9	(1)	2	—	11	(1)
CNNet	8	(1)	3	—	11	(1)
GP	4	—	8	(1)	12	(1)
FNNet	0	—	5	—	5	—

Table 5.15. *Diagnostics from fitting a mixed-effects model for analyzing the performance of various model architectures.*

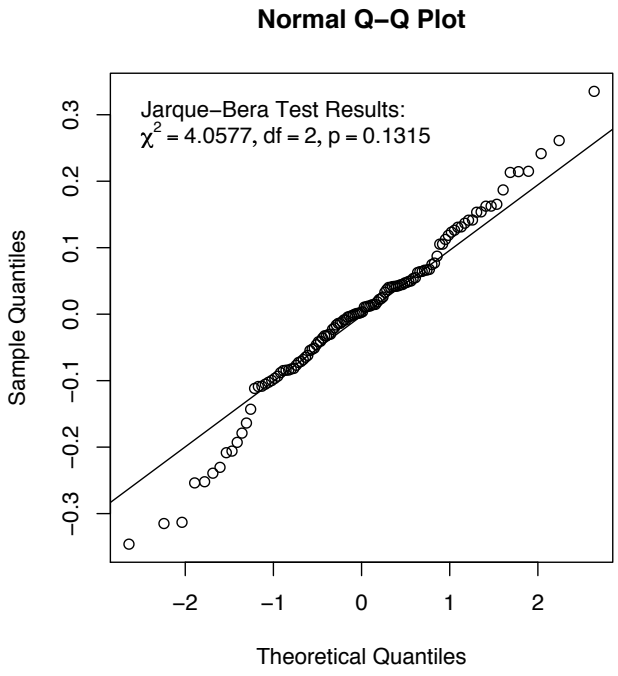
```
Fixed effects: Sharpe ~ Controller
                                       Value    Std.Error   DF    t-value   p-value
(Intercept)                       0.21672104   0.04616619   92   4.694367    0.0000
Controller = FinancialNNet        0.07055642   0.03907024   92   1.805887    0.0742
Controller = GaussianProcess      0.00067179   0.03907024   92   0.017194    0.9863
Controller = LinearRegressor      0.02960146   0.03907024   92   0.757647    0.4506
Controller = LongOnly             0.01610538   0.03907024   92   0.412216    0.6811

Correlation:
                                (Intr) CntFNN CntrGP CntrLR
Controller = FinancialNNet      -0.423
Controller = GaussianProcess    -0.423  0.500
Controller = LinearRegressor    -0.423  0.500  0.500
Controller = LongOnly           -0.423  0.500  0.500  0.500

Random effects:
 Formula: ~1 | Identifier
        (Intercept)  Residual
StdDev:   0.1812010 0.1353433

Standardized Within-Group Residuals:
        Min          Q1         Med          Q3         Max
-2.55597583 -0.50948698  0.02186522  0.47267918  2.47665678

Number of Observations: 120
Number of Groups: 24
```



◀ **Figure 5.32.** *Q-Q plot of the residuals of the linear mixed-effects model against a theoretical normal distribution. The Jarque-Bera test does not reject the normal null hypothesis.*

does bring tangible value, on average, against alternative models (including a buy-and-hold default model), after controlling for the random effects of the market and time period.

5.8 Discussion and Future Work

The current results, although demonstrating the value of the proposed algorithm, perhaps raise more questions than they answer. In particular, we ignored the major efficiency gains that can be achieved by an intelligent pruning of the search graph, either in the form of beam searching or action enumeration heuristics. The following conclusions can be drawn:

On Optimizing Sharpe Ratios In The Presence Of Transaction Costs

- We noted that the Sharpe ratio, despite some criticism, is an adequate criterion for *comparing* sequences of realized returns. However, it utterly fails as an *optimization objective* and must be augmented with regularization and risk preferences.
- We observed that pure in-sample optimization of Sharpe ratios is a difficult problem in the presence of transaction costs due to the prevalence of local minima. Gradient-based approaches progressively become completely ineffective beyond a low level of transaction costs (25

basis points). This is analogous to the behavior of Ising models in solid-state physics (MacKay 2003).

- Solutions based on discrete enumeration of actions are shown to be effective (K-best paths).

On Modeling

- We showed the value of training a learning algorithm with a financial criterion, but also the crucial need to regularize it for good out-of-sample performance (and noted, in passing, that this need for regularization may be related to the behavior of the Hessian around the optimum).

- Prefitting using a traditional classification objective helps considerably, in part because it breaks recurrence and it constitutes an "easier" problem.

- It is not obvious that using K-Best targets for prefitting or within the "hint term" in the main optimization objective brings about better performance than simpler approaches such as the monthly or daily return.

- Considering other learning algorithms, we clearly observe that *simpler is better* (in the sense of more constrained or more regularized models). Even purportedly automatically-regularized models such as Gaussian processes generally show disappointing performance for this task.

- We also note the value of training according to a financial criterion, or of principles that acknowledge the ordered class structure.

Future Work

The mixed results of §5.7/p. 173 make one wonder whether economic causes could explain some of the observed behavior. At this juncture, we can more speculate than provide firm answers. For instance, the Gaussian process model is found to be very poor in the US market, but quite good otherwise: in the first case (US performance), this is attributable to the severe overfitting that the GP exhibits, wherein the model systematically fits nearly perfectly on the training set, but is incapable to generalize out of sample. For the Canadian and European market, could the GP's good performance be exploiting inefficiencies that would more likely arise in these markets than the widely-followed US market? This should be the subject for further study.

Likewise, the Linear Regression model significantly underperforms in the European market, whereas the Financial and Classification neural networks perform well. This is puzzling, given that all three models comprise almost

the same number of free parameters and carry out the same linear projection operation to input variables; they differ only in their output layer and training criterion. Here, the investigation should focus on whether structural breaks in European series—possibly caused by the introduction of the Euro—could have more effect on the OLS objective than the others. One possible test would be to experiment with robust regression methods, that are less sensitive to outliers than OLS.

Additional topics for further study also include:

- The current "best-performing" model is but a simple modification of the proportional-odds model: made up of a linear combination of inputs, followed by a sequence of ordered classifiers to make the final decision. In particular, the "controller" is not one in the traditional sense of dynamic programming, since it does not even use the current portfolio state as input.* This is ironic, since the K-best approach does provide "perfect state variables" with which to train the model in a supervised fashion, and one is not faced with the long simulation-based procedures of approximate dynamic programming (§3.2/p. 77). Although we experimented with the latter form, the results have been disappointing so far, very probably since only a single historical trajectory was provided as part of the training set. To overcome the history-richness issue, one can envision several possibilities:

  1. State perturbation: at each time-step, a perturbation of the portfolio can be made, wherein the portfolio is forced to take on specific values. Recomputing the optimal action under such a constraint induces changes in the portfolios at neighboring time-steps with respect to the baseline of having no constraint: these changes are added to the training set to increase the state diversity. Depending on the level of transaction costs, the impact of perturbation constraints can be localized (for low costs) or far-ranging (at high costs). This approach is obviously complicated by the need to simulate the type of perturbations that might realistically occur.
  2. Using several assets. If one deals with individual stocks which share the same input variables, the training objective can obviously incorporate historical data from all stocks. This can be memory-intensive if all data is required to be held in memory during optimization.
  3. Block bootstrap: synthesize several histories if we have only a single asset. This suffers from the same memory problem as previously, and in addition, one must be careful that the bootstrap distribution matches the empirical one (Lahiri 2003).

*In this context, the effect of incorporating transaction costs in the objective are then to downweight the impact of variables, particularly the technical ones, that would incur excessive trading activity.

- Portfolio of assets. There are issues of scaling with the number of assets, and the effectiveness of the beam-searching and action-enumeration heuristics remains to be evaluated.

- Mixtures of models, of which bagging (Breiman 1996) is a simple example, could be of help for the selection of hyperparameters, as was done by Chapados (2000).

5.A Appendix: Statistical Techniques

We outline in this appendix a number of methodological tools, use of which is made in this chapter to analyze experimental results.

5.A.1 Bootstrap Inference for the Sharpe Ratio

Given two distinct investment strategies yielding, respectively, sets of realized returns $r_{t,1}$ and $r_{t,2}, t = 1, \ldots, T$, one could naturally seek to test the hypothesis of whether the difference between the two measured Sharpe ratios is statistically significant. A number of tests have been proposed in the literature, starting with Jobson and Korkie (1981), with improvements by Memmel (2003). However, these tests tend to be too liberal for financial data (i.e. reject the null hypothesis more often than their nominal size) since they assume that the returns are IID and/or normally distributed, two assumptions known to be violated in financial time series. We summarize in this section a recent test based on the studentized block bootstrap proposed by Ledoit and Wolf (2008), which is robust to violations of both conditions.

Assume underlying stationary return distributions over the complete time period, with associated respective means and variances denoted by μ_1, μ_2, σ_1^2, and σ_2^2. Let the sample means and variances be denoted by $\hat{\mu}_1, \hat{\mu}_2, \hat{\sigma}_1^2$, and $\hat{\sigma}_2^2$. The "true" difference between the Sharpe ratios is given by

$$\Delta = \frac{\mu_1}{\sigma_1} - \frac{\mu_2}{\sigma_2},$$

and a sample estimate is obtained as

$$\hat{\Delta} = \frac{\hat{\mu}_1}{\hat{\sigma}_1} - \frac{\hat{\mu}_2}{\hat{\sigma}_2}.$$

For notational simplicity, denote the uncentered second moments as $\gamma_1 \equiv \mathbb{E}[r_{t,1}^2]$ and $\gamma_2 \equiv \mathbb{E}[r_{t,2}^2]$, and their sample estimates as $\hat{\gamma}_1$ and $\hat{\gamma}_2$. We also write the vector of all parameters and estimates as

$$\upsilon = (\mu_1, \mu_2, \gamma_1, \gamma_2)' \qquad \text{and} \qquad \hat{\upsilon} = (\hat{\mu}_1, \hat{\mu}_2, \hat{\gamma}_1, \hat{\gamma}_2)'.$$

The Sharpe ratio difference can therefore be written as

$$\Delta(\upsilon) = f(a,b,c,d) = \frac{a}{\sqrt{c-a^2}} - \frac{b}{\sqrt{d-b^2}}.$$

From the Central Limit Theorem, we can assume, under some regularity conditions (e.g. Andrews 1991, or Lahiri 2003), that

$$\sqrt{T}(\hat{\upsilon} - \upsilon) \xrightarrow{d} N(0, \boldsymbol{\Psi}),$$

where $\boldsymbol{\Psi}$ is some (unknown) symmetric non-negative definite covariance matrix between the parameters. From the delta method (Greene 2007), we obtain the asymptotic distribution of the Sharpe ratio difference as*

$$\sqrt{T}(\hat{\Delta} - \Delta) \xrightarrow{d} N\big(0, \nabla' f(\upsilon)\, \boldsymbol{\Psi}\, \nabla f(\upsilon)\big),$$

where $\nabla f(\cdot)$ is the gradient vector of f,

$$\nabla f(a,b,c,d) = \Big(\frac{c}{(c-a^2)^{3/2}}, -\frac{d}{(d-b^2)^{3/2}}, -\frac{a}{2(c-a^2)^{3/2}}, \frac{b}{2(d-b^2)^{3/2}}\Big)'.$$

Given a consistent estimator of $\boldsymbol{\Psi}$, denoted $\hat{\boldsymbol{\Psi}}$, an asymptotic standard error for $\hat{\Delta}$ is obtained as

$$s(\hat{\Delta}) = \sqrt{\frac{\nabla' f(\hat{\upsilon})\, \hat{\boldsymbol{\Psi}}\, \nabla f(\hat{\upsilon})}{T}}.$$

The estimator $\hat{\boldsymbol{\Psi}}$ is computed via bootstrap resampling using the circular block bootstrap of Politis and Romano (1992) and the prewhitened Quadratic-Spectral kernel of Andrews and Monahan (1992). In all our tests, the block size was fixed at 5, although Ledoit and Wolf (2008) present a (computationally-intensive) technique to calibrate this parameter. (The value of 5 was, on average, good in Ledoit and Wolf's experiments.)

From a computed standard error, a p-value for the original studentized test statistic

$$d = \frac{|\hat{\Delta}|}{s(\hat{\Delta})}$$

can be obtained as follows. Denote the centered studentized statistic computed from the m-th bootstrap sample by $\tilde{d}^*_m$, $m = 1, \ldots, M$ (where M is the total number of bootstrap resamples),

$$\tilde{d}^*_m = \frac{|\hat{\Delta}^*_m - \hat{\Delta}|}{s(\hat{\Delta}^*_m)},$$

with Δ^*_m the m-th bootstrap Sharpe ratio difference. We obtain the p-value of $\hat{\Delta}$ as

$$p = \frac{1}{M+1}\Big(1 + \sum_{m=1}^{M} I[\tilde{d}^*_m \geq d]\Big),$$

where $I[\cdot]$ is the indicator function.

*The result of the delta method is easily obtained by considering a Taylor series expansion of $\hat{\upsilon}$ around υ.

5.A.2 Analysis of Variance

*Analysis of Variance.

The ANOVA* is a general statistical technique that is used to decompose the variability of a measured variable in terms of explanatory variables called *factors* (or, oftentimes, *treatments*). It is a widely-used procedure in the analysis of experimental results: in this context the factors are generally discrete variables and correspond to distinct experimental conditions. In our case, we shall use ANOVAs to understand how the observed test performance of a model is affected by the levels of hyperparameters when those are varied within the experiment. The analysis of variance (and the related design of experiments) is a large topic; see, e.g., Box, Hunter, and Hunter (2005) for more information.†

The fundamental purpose of an ANOVA is to test hypotheses about equality of means. In the simplest setting, suppose that we perform an experiment by varying a single factor: we are interested in determining whether variations in this factor, at different *experimental levels*, have a significant impact on measured performance. Formulated as a classical inference problem, we want to test the null hypothesis that the *mean performance* is equal for all levels of the factor. A rejection of the null hypothesis indicates that there are some levels for which performance is significantly different from others.‡

Consider again a single factor which can take on one of M levels, and assume that for each level, we perform N trials each having a random outcome. Then we can model the result of trial j at level i of the factor as

$$y_{ij} = \mu + \beta_i + \varepsilon_{ij}, \qquad i = 1, \ldots, M, j = 1, \ldots, N, \tag{5.32}$$

§INTERACTION EFFECT: *Combined effect of two or more independent variables on a dependent variable when their joint effect is not additive. Used in analysis of variance tables as a correction to* main effects *to account for nonlinearities.*

where μ is the "grand mean" (overall average performance), β_i is the average deviation from the grand mean due to the effect of factor i, and ε_{ij} a residual. This gives rise to a test of the null hypothesis that all β_i's are zero, namely that the factors have no impact on performance. In a classical ANOVA, this is performed by a variance ratio test (F-test) that compares the variance among the β_i's to the variance of the residuals, and concludes in favor of the null if the two are not statistically different (whence the name "analysis of *variance*).

With more than one factor, the procedure remains the same but can now incorporate *interaction effects*§, in addition to the *main effects.*¶ For instance, with two factors, the result of the k-th trial at level i of the first factor and level j of the second is modeled as

$$y_{ijk} = \mu + \beta_i + \gamma_j + (\beta : \gamma)_{ij} + \varepsilon_{ijk},$$

¶MAIN EFFECT: *Effect of an independent variable (hyperparameter) on a dependent variable (e.g. classification accuracy), averaging over all levels of the other independent variables. Contrast with* interaction effect.

where β_i represents the main effect for the first factor, γ_j the main effect for the second, and $(\beta : \gamma)_{ij}$ is an interaction term modeling the additional

†The excellent free online handbook edited by Croarkin and Tobias (2006) and maintained by the U.S. National Institute of Standards and Technology contains a wealth of information; section 7.4.3 explains ANOVAs in detail.

‡But it does not tell us *which* levels are "better"; this necessitates the use of so-called *post hoc* analyses, such as the Tukey pairwise comparison procedure described next.

impact of levels i and j occurring together. If no interaction is suspected, this term is omitted, but it is common to explicitly test for the significance of interactions before omitting them. The inference proceeds exactly as before, through a series of F-tests.

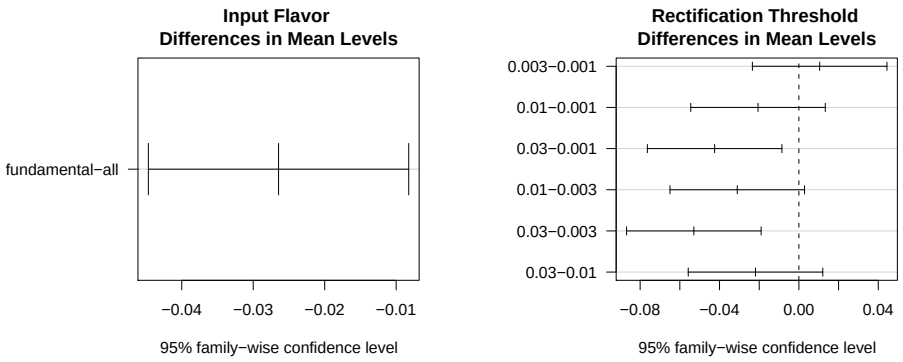
In this chapter, all ANOVA tables are presented along the following lines:

```
                               Df    Sum Sq   Mean Sq   F value    Pr(>F)
Date                            3    5.5900    1.8633  688.3642 < 2.2e-16 ***
EquityIndexes.input.flavor      1    0.0224    0.0224    8.2732  0.004782 **
EquityIndexes.rectify.thresh    3    0.0526    0.0175    6.4829  0.000426 ***
OptLR.weight.decay              3 3.988e-09 1.329e-09 4.911e-07  1.000000
Residuals                     117    0.3167    0.0027
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Here, we report the results of an experiment (a linear regressor model on the US market, for the period ending in 1998; see §5.B.3/p. 199) with four factors: the time period (`Date`), type of input variables used (fundamental or a combination of fundamental and technical; see §5.B/p. 197), the rectification thresholds for converting a continuous output in a decision and the weight decay for linear regression. There are no tests for interaction effects between factors (but we would abbreviate them by the mention `Ix` in the table). The column `Df` in the ANOVA table stands for "degrees of freedom", and is one less than the number of levels in each factor (for main effects only). For example, the `Date` factor has four distinct levels (each corresponding to an approximately two-year timespan), and thus three degrees of freedom. The `Sum Sq` column is the sum of squared deviations from the grand mean absorbed by the factor, and `Mean Sq` is `Sum Sq` divided by `Df`. The column `F value` is the value of the variance ratio statistic, namely the `Mean Sq` of the factor being tested to the `Mean Sq` of the residuals. Finally, the column `Pr(>f)` is the p-value of observing such an extreme variance ratio, using Fisher's F distribution with the factor's degrees of freedom in the numerator, and the residuals' degrees of freedom in the denominator.

What this table tells us is the first three main effects have a highly significant impact on performance (both their p-values are smaller than conventionally-accepted thresholds such as $p = 0.05$), but the fourth one (weight decay) does not. Put differently, the performance can be approximated by the sum of the first three main effects, without a need to account for an additional term correcting the last effect.

Occasionally, higher-order interactions (greater than degree two) are necessary to represent complex empirical relationships between hyperparameters. Mathematically, they are a straightforward generalization of order-two interactions: whereas the latter correct for effects that are predicted by separately considering two individual variables, an order-three interaction, say, corrects for effects that are predicted by all individual order-one and order-two terms.



▶ **Figure 5.33.** *Graphical result of Tukey's* ***honestly significant differences*** *procedure. It presents the results of all pairwise mean differences between the levels of factors involved in an experiment, and associated confidence intervals on this difference.*

5.A.3 Tukey's Adjustment for Multiple Comparisons

The ANOVA is helpful to understand which factors are driving performance; however, it only tells us that, e.g. the rectification threshold is significant, without telling us *which level* of a given factor produces the best performance (assuming no interactions). This is the purpose of secondary analyses, of which Tukey's method of *honestly significant differences* is used extensively in this chapter.

Consider the β_i's arising from the one-factor ANOVA (5.32). Obviously, one can look at the "largest" β to determine the factor level yielding the highest performance, on average. But what about the "second largest" β: do we have evidence to affirm that it is really worse than the best?

One way to answer this question is to perform all pairwise comparisons of means and, for each comparison, test the hypothesis of whether the two means are significantly different. One could imagine repeating a series of t-tests to carry out this task, but the problem is that all those tests *are not independent* (since they involve the same mean several times). As such, repeated t-tests will have an effective size larger than the nominal size, increasing the possibility of type-I error.

Tukey's method of *honestly significant differences* (Hsu 1996) is a way to perform all pairwise mean comparisons in a single step, and control for experiment-wise error rate (also called the "family error rate"). Without going into details of mathematical procedure, the approach is very similar to repeated t-tests, except that it corrects for the occurrence of multiple comparisons.

In this chapter, we present the results of Tukey's test in the form of plots of all pairwise mean differences between the levels of the factors involved in the experiment. An example corresponding the the previous ANOVA table appears in Fig. 5.33 (repeated from Fig. 5.34). From the figure, we note, for instance, that making use of only fundamental variables (the "fundamental–all" entry in the left pane) decreases performance compared to using all variables (the mean difference is negative and significant). From the right panel, we see that using a rectification threshold of 0.03 would be a bad

choice whereas there is no significant differences between 0.003 and 0.001.

When interactions must be displayed, the plots become more crowded since there are generally a large number of possible pairwise comparisons between interacting factors. For clarity, such plots are sometimes omitted from the results presented herein, but we do take their results into account when performing, for example, model selection.

5.B Appendix: Input Variables

The utilized input variables are country-specific,* but not model-specific. In other words, all models compared within a given country share the same inputs, and no attempt was made to perform model-specific variable selection to tune the performance of a specific model. We describe each country's variables in detail below.

In all cases, the input variables fall into two categories: the first, *fundamental variables*, reflect broad macroeconomic conditions. These variables vary relatively slowly, typically monthly or quarterly; since they are quite country-specific, they are described in detail in the following country subsections. The second category of inputs are so-called *technical variables*, which attempt to capture short-term price patterns and likely market direction following specific combination of indicators. In a sense, they are reflective of market psychology and can be somewhat predictive at short horizons (a few days, at most), although their significance is highly market- and time-period-dependent.

5.B.1 Technical

The technical variables are strongly inspired by the work of Collins (2006), which specializes on U.S. equity, commodity and foreign exchange futures. The following variables are used for all countries, although not all variables are used for any given country. All variables are binary indicators, taking on value 1 if the condition is satisfied, and 0 otherwise. (The prefix `l` in the variable names indicates that they are suggestive of a long bias, according to Collins; however, the models are trained without imposing any constraint—sign or otherwise—on model coefficients.)

Variable Name	Description
`l.momentum`	The close is greater than the 40-day closing average.
`l.reversal`	The two-day close is less than the five-day close.
`l.extremaorder`	The highest high of the last 50 days occurs before the lowest low of the last 50 days.

*Note that for terminological simplicity, we consider "Europe" to be a country, in addition to the U.S. and Canada; it is hoped that European readers will not have sensibilities hurt by this egregious abuse of language.

`l.range`	The range is smaller than the 10-day average range and the close is higher –**OR**– the range is bigger than the 10-day range and the close is lower. (Smaller ranges mean 'go-with', namely continue in the same direction; bigger mean 'fade', i.e. take a contrarian stance).
`l.midpoint`	The close is higher than the average 15-day midpoints (15-day highs plus 15-day lows divided by 2).
`l.conseqclose`	The close was less than the open two days in a row.
`l.cup`	Yesterday completed a cup formation.
`l.congestion`	Yesterday completed a three-bar congestion of lows: the highest low minus lowest low being equal to or less than 20 percent of the combined three-day range.

Some variables are used with a `s.` prefix, which mostly reverses the indication given by the corresponding `l.` variable. Some biases are induced from calendar events. The variables for calendar rules are defined as follows:*

1. Day of week (variable `l.dow`)):
   - Monday: buy or sell in the direction of Friday's close-to-close net change
   - Tuesday: go opposite of Monday's close-to-close net change
   - Wednesday–Friday: fade (i.e. go opposite) the largest open-close move that happened earlier during the week
2. Day of the month (variable `l.dom`):
   - Go short from the 7th until the 22nd day of the month
   - Go long otherwise
3. Month of the year (variable `l.month`):
   - Go long from November 2nd until May 1st
   - Go short otherwise

5.B.2 Notes on Time-Series Regressions

In the following pages, we produce regression results to illustrate the forecasting power of country-specific fundamental and technical input variables on equity returns (monthly and daily returns are considered separately). For this purpose, we use a standard OLS regression, whose goal is not to provide the "best" model, but just to assess the significance of the variables under consideration.

However, one must use care when using regressions for this purpose, as it is well-known that autocorrelations and heteroskedasticity in dependent variables (here, equity returns) will produce optimistically-biased standard

*"Going long" means taking on a value of one for these variables; "going short" corresponds to a value of zero.

errors on regression coefficients (Greene 2007) (i.e. the p-values will be too small, rejecting the null hypothesis more often than the nominal size of the test). To correct for this, all linear regression standard errors (and associated p-values) presented below are adjusted using the Andrews (1991) heteroskedasticity- and autocorrelation-consistent estimator of the covariance matrix of model parameters.* This correction makes p-values more trustworthy, particularly in the daily models where (as is shown below) there is significant autocorrelation in the OLS residuals.

All regression results are presented in the following format:

- Description of model coefficients, estimates, standard errors, t-statistics and associated p-values; as noted, the Andrews (1991) estimator is used in computations.
- In-sample goodness of fit statistics (R^2 and adjusted-R^2).
- Results from a Ljung and Box (1978) test to evaluate the autocorrelation in the residuals (tested with a lag of 10). A small p-value is an indication to reject the null hypothesis of absence of autocorrelation in the residuals.
- Normal Q-Q plot and autocorrelogram of the residuals.

Notation for Regression Coefficients

To ease presentation, we use a time-series notation inspired by the R statistical modeling language (R Development Core Team 2008)† to show the transformations carried out on each input variable before it is used within the regression. The following elements are used:

- $d(X, n)$: time difference of variable X with lag n, $X_t - X_{t-n}$.
- $\ell(X, -n)$: value of variable X taken n time-steps in the past, X_{t-n}.

5.B.3 United States

Fundamental Variables

The following raw variables are used within U.S. models:

*The Andrews estimator is a generalization of the well-known Newey and West (1987) estimator. The number of lags in the estimator is chosen adaptively and Andrews' "quadratic spectral" kernel is used, as implemented by the `weightsAndrews` function in the R **sandwich** package (Zeileis 2004).

†More specifically, the `dyn` package (Grothendieck 2005), which extends R's standard modeling notation to dynamical models.

Variable Name	Description
`sp500_spot`	S&P 500 spot price
`yield_10yr`	Ten-year U.S. government bond yield
`yield_3mo`	Three-month U.S. T-bills yield
`sp500_anndiv`	Annualized dividend of the S&P 500 constituents
`sp500_annearn`	Annualized earnings of the S&P 500 constituents
`cpi`	Consumer Price Index (U.S.)
`vix`	CBOE Volatility index (revised 2003 methodology, with CBOE data backfill to 1986)
`crude_spot`	Spot price of crude oil
`leading_indic`	Conference Board Index of Leading Economic Indicators (U.S.)
`presyear.3`	1 if in third year of the 4-year U.S. presidential cycle (dummy variable); the effects of the four-year U.S. Presidential Cycle on equity indices are documented by Fisher (2006).

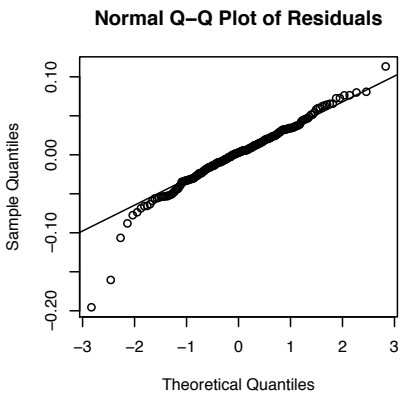
S&P 500 Monthly Returns

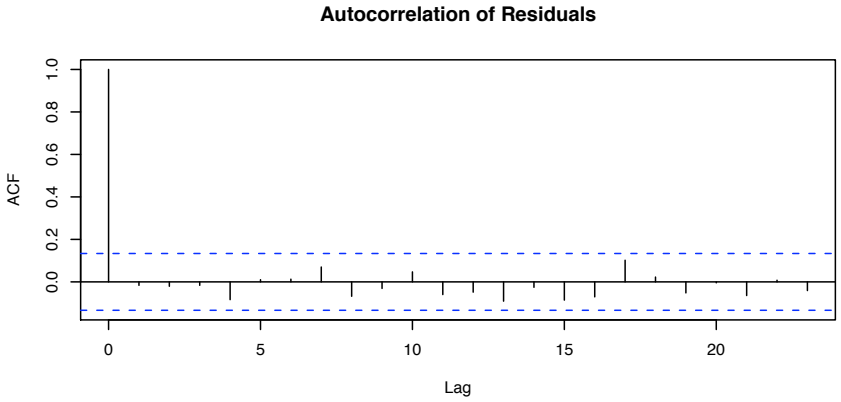
Table 5.19 shows the results of a linear regression of monthly S&P 500 returns (price index only) on a set of inputs derived from the fundamental variables listed previously. The purpose of this exercise is only to assess the explanatory power of the fundamental variables and their eventual utility in conjunction with the models compared in §5.6.1/p. 169. Monthly data over the period 1987–2004 is used in the regression (215 observations in dataset; due to long lags in some variables, raw data starts in January 1983). The following input variable transformations are provided:

Transformation Notation	Description
$d(\log(\texttt{sp500\_spot}), 1)$	One-month stocks log-return
$d(\log(\texttt{sp500\_spot}), 48)$	Four-year stocks log-return
$d(\texttt{yield\_10yr} - \texttt{yield\_3mo}, 3)$	Three-month increase in term spread (difference between long and short rates)
$d(\texttt{sp500\_anndiv}/\texttt{sp500\_spot}), 24)$	Two-year increase in stock dividend yield
$\ell(d(\texttt{sp500\_anndiv}/\texttt{sp500\_spot}, 24), -24)$	Two-year increase in stock dividend yield (lagged by two years)
$\texttt{sp500\_annearn}/\texttt{sp500\_spot}$	Stock earnings yield
$d(\texttt{sp500\_annearn}/\texttt{sp500\_spot}, 24)$	Increase in stock earnings yield over the past two years
$d(\log(\texttt{cpi}), 12)$	Log consumer prices increase over the previous year (inflation measure)
$d(\texttt{vix}, 1)$	One-month increase in stock market volatility
$d(\log(\texttt{crude\_spot}), 1)$	One-month increase in (log) crude oil prices
$d(\log(\texttt{leading\_indic}), 1)$	One month increase in (log) LEI.

Table 5.19. *Linear regression of monthly S&P 500 returns using U.S. input variables.*

Coefficient	Estimate	Std Err	t Value	$\Pr(>\|t\|)$	
$d(\log(\texttt{sp500\_spot}), 1)$	−0.15258987	0.08382644	−1.8203	0.0701922	.
$d(\log(\texttt{sp500\_spot}), 48)$	0.08084089	0.03907743	2.0687	0.0398440	*
`yield_10yr`	−0.01538950	0.00499746	−3.0795	0.0023621	**
$d(\texttt{yield\_10yr} - \texttt{yield\_3mo}, 3)$	0.00889335	0.00510698	1.7414	0.0831335	.
$d(\texttt{sp500\_anndiv/sp500\_spot}), 24)$	4.51215181	2.74706045	1.6425	0.1020343	
$\ell(d(\texttt{sp500\_anndiv/sp500\_spot}, 24), -24)$	3.38139796	1.80857491	1.8696	0.0629797	.
`sp500_annearn/sp500_spot`	3.03631204	0.75775450	4.0070	$8.645e-05$	***
$d(\texttt{sp500\_annearn/sp500\_spot}, 24)$	−0.67193203	0.31946096	−2.1033	0.0366748	*
$d(\log(\texttt{cpi}), 12)$	−1.79144661	0.69653860	−2.5719	0.0108314	*
$d(\texttt{vix}, 1)$	−0.00216637	0.00055049	−3.9353	0.0001143	***
$d(\log(\texttt{crude\_spot}), 1)$	−0.09575573	0.03009973	−3.1813	0.0016975	**
$d(\log(\texttt{leading\_indic}), 1)$	−0.78695825	0.52876859	−1.4883	0.1382353	
`presyear.3`	0.01411595	0.00568320	2.4838	0.0138125	*
R^2 : 0.2506 Adjusted R^2 : 0.2024	Box-Ljung test results: $\chi^2 = 4.6691, df = 10, p = 0.9122$				





S&P 500 Daily Returns

Table 5.20 shows the results of a linear regression of daily S&P 500 returns (price index) on inputs similar to those previously described for the monthly-returns model. Daily data over the period 1986-10-08 to 2004-12-30 is used in the regression (due to lagged inputs, raw data starts on 1983-01-03). Note that all lags are now expressed in days. In addition to the fundamental variables, some technical indicators described in §5.B.1/p. 197 are also used. The Box-Ljung test shows that significant autocorrelation of residuals is observed at the daily level; this causes an upwards adjustment to the p-value of most coefficients compared to plain OLS. Despite this, many coefficients that were significant at the monthly level remain so at the daily horizon (even though the target variable is substantially different). Moreover, several technical indicators show statistical significance.

5.B.4 Canada

Fundamental Variables

The following raw variables are used within Canadian models:

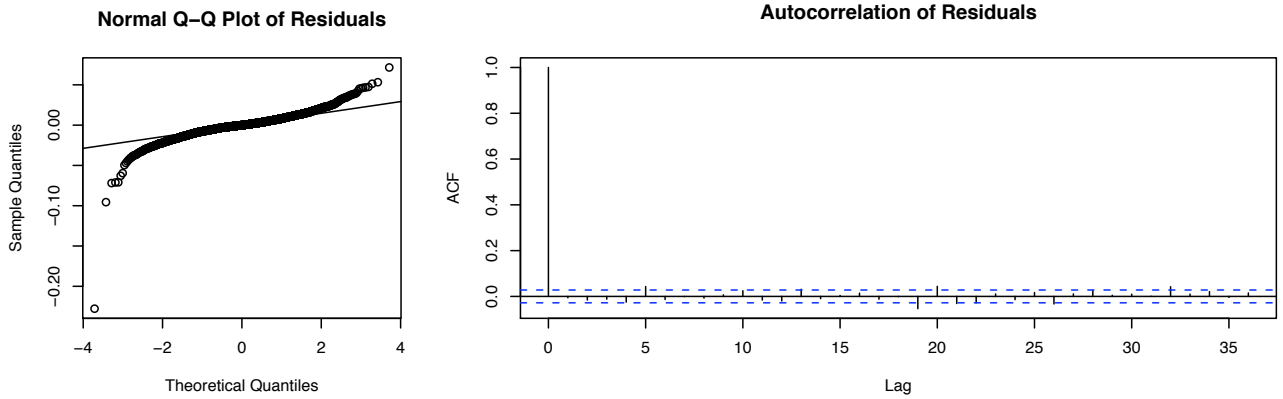
Variable Name	Description
`tsx_spot`	Spot level of the TSX-Composite Price Index; before May 1st 2002, this is the TSE-300 index
`ca_yield_10yr`	Ten-year Canadian government bond yield
`ca_yield_3mo`	Three-month Canadian government bond yield
`tsx_div_yld`	Dividend yield of the TSX-Composite index
`tsx_earn_yld`	Earnings yield of the TSX-Composite index
`cpi`	Consumer Price Index (Canada)
`mvx`	Montreal Exchange Implied Volatility Index
`crude_spot_cad`	Spot price of crude oil (in Canadian dollars)
`gold_cad`	Spot price of gold (in Canadian dollars)
`silver_cad`	Spot price of silver (in Canadian dollars)
`gsci_cad`	Goldman Sachs Commodity Index (in Canadian dollars)
`leading_indic`	Conference Board Index of Leading Economic Indicators (Canada)
`exchange_rate`	CAD/USD exchange rate
`unemployment`	Total unemployment rate (seasonally adjusted)

TSX Monthly Returns

Table 5.23 shows the results of a linear regression of monthly TSX-Composite returns (price index) on a set of inputs derived from the fundamental variables listed previously. Monthly data on the period February 1986–December 2004 are used in the regression, on monthly returns (211 observations in dataset instead of 226, due to some missing values; due to long lags in some variables, raw data starts in January 1982). The following input vari-

Table 5.20. *Linear regression of daily S&P 500 returns using U.S. input variables.*

Coefficient	Estimate	Std Err	t Value	$\Pr(>\|t\|)$	
$d(\log(\texttt{sp500\_spot}), 1)$	0.04222557	0.04648483	0.9084	0.363726	
$d(\log(\texttt{sp500\_spot}), 63)$	−0.00511318	0.00323051	−1.5828	0.113537	
$d(\log(\texttt{sp500\_spot}), 4 * 252)$	0.00248588	0.00077681	3.2001	0.001383	**
`yield_10yr`	−0.00137013	0.00031979	−4.2845	$1.867e-05$	***
$d(\texttt{yield\_10yr} - \texttt{yield\_3mo}, 63)$	0.00058057	0.00036850	1.5755	0.115211	
`sp500_anndiv/sp500_spot`	0.11083306	0.04140145	2.6770	0.007453	**
`sp500_annearn/sp500_spot`	0.08420164	0.02101269	4.0072	$6.236e-05$	***
$d(\log(\texttt{cpi}), 1)$	0.56307393	0.17362142	3.2431	0.001190	**
$d(\texttt{vix}, 1)$	0.00029173	0.00014106	2.0682	0.038677	*
$d(\texttt{vix}, 10)$	0.00012112	0.00010129	1.1958	0.231836	
$d(\log(\texttt{crude\_spot}), 21)$	−0.00240148	0.00182062	−1.3190	0.187216	
$d(\log(\texttt{leading\_indic}), 1)$	0.34665267	0.13318594	2.6028	0.009275	**
`l_cup`	0.00053254	0.00054270	0.9813	0.326507	
`l_dom`	0.00052313	0.00032452	1.6120	0.107022	
`l_dow`	0.00105747	0.00040268	2.6261	0.008664	**
`l_midpoint`	0.00099028	0.00038524	2.5705	0.010184	*
`l_month`	0.00041302	0.00028883	1.4300	0.152784	
`s_cap`	−0.00121878	0.00059657	−2.0430	0.041109	*
`presyear.3`	0.00082553	0.00036697	2.2496	0.024520	*
$R^2 : 0.02574$ Adjusted $R^2 : 0.02194$	Box-Ljung test results: $\chi^2 = 18.6893, df = 10, p = 0.04439$				



able transformations are provided to the monthly model; note that since the Canadian economy is heavily dependent on natural resources, the price variation of important commodities turns out to have an impact on the stock market performance:

Transformation Notation	Description
$d(\log(\texttt{tsx\_spot}), 6)$	Six-month stocks log-return
$d(\log(\texttt{tsx\_spot}), 12)$	One-year stocks log-return
$d(\log(\texttt{tsx\_spot}), 24)$	Two-year stocks log-return
$d(\log(\texttt{tsx\_spot}), 48)$	Four-year stocks log-return
$\texttt{ca\_yield\_10yr}$	Ten-year Canadian government bond yield
$\texttt{ca\_yield\_10yr} - \texttt{ca\_yield\_3mo}$	Canadian term spread
$d(\texttt{ca\_yield\_10yr} - \texttt{ca\_yield\_3mo}, 24)$	Two-year increase in term spread
$\texttt{ca\_yield\_3mo} - \texttt{us\_yield\_3mo}$	Canadian sovereign spread (difference between Canadian and U.S. short rates)
$d(\texttt{tsx\_div\_yld}, 24)$	Two-year increase in dividend yield
$d(\log(\texttt{cpi}), 12)$	Log consumer prices increase over the previous year (inflation measure)
$d(\texttt{mvx}, 1)$	One-month increase in stock market volatility
$d(\log(\texttt{crude\_spot\_cad}), 1)$	One-month increase in crude oil price
$d(\log(\texttt{gold\_cad}), 3)$	Three-month increase in gold price
$d(\log(\texttt{silver\_cad}), 12)$	Twelve-month increase in silver price
$d(\log(\texttt{gsci\_cad}), 12)$	Twelve-month increase in a major commodity index (heavily weighted towards energy commodities)
$d(\log(\texttt{leading\_indic}), 1)$	One-month increase in (log) LEI
$d(\texttt{exchange\_rate}, 12)$	Twelve-month variation in exchange rate
$d(\log(\texttt{unemployment}), 12)$	Twelve-month variation in (log) unemployment

TSX Daily Returns

Table 5.24 shows the results of a linear regression of daily TSX-Composite returns (price index) on inputs similar to those previously described for the monthly-returns model. Daily data over the period 1986-12-11 to 2004-12-30 is used in the regression. Note that all lags are now expressed in days. In addition to the fundamental variables, some technical indicators described in §5.B.1/p. 197 are also used. The same high autocorrelation of residuals observed for daily U.S. returns is also seen in the Canadian case.

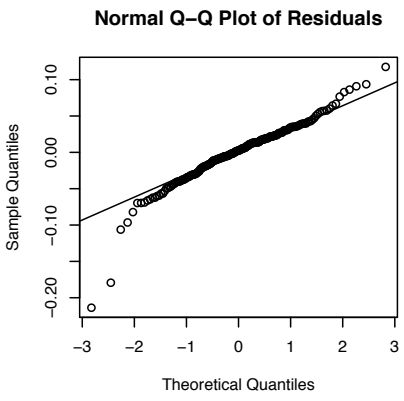
5.B.5 Europe

Fundamental Variables

The handling of Europe is slightly more involved than the U.S. and Canada, since it is made up of a number of important economies, and the introduction of the Euro constitutes a large break in the historical data series. We

Table 5.23. *Linear regression of monthly TSX-Composite returns using Canadian input variables.*

Coefficient	Estimate	Std Err	t Value	$\Pr(>\|t\|)$	
$d(\log(\texttt{tsx\_spot}), 6)$	−0.04963826	0.04844755	−1.0246	0.3068586	
$d(\log(\texttt{tsx\_spot}), 12)$	0.03380088	0.03721920	0.9082	0.3649395	
$d(\log(\texttt{tsx\_spot}), 24)$	−0.12589037	0.04724693	−2.6645	0.0083692	**
$d(\log(\texttt{tsx\_spot}), 48)$	−0.08476092	0.03021892	−2.8049	0.0055539	**
`ca_yield_10yr`	−0.00561855	0.00705629	−0.7962	0.4268769	
$d(\texttt{ca\_yield\_10yr} - \texttt{ca\_yield\_3mo}, 24)$	−0.00469095	0.00274042	−1.7118	0.0885632	.
`ca_yield_10yr` − `ca_yield_3mo`	−0.00656029	0.00391823	−1.6743	0.0957086	.
`ca_yield_3mo` − `us_yield_3mo`	−0.02185812	0.00521051	−4.1950	$4.175e-05$	***
`tsx_div_yld`	4.22539440	2.22219878	1.9014	0.0587488	.
$d(\texttt{tsx\_div\_yld}, 24)$	−4.57769347	2.15955082	−2.1197	0.0353190	*
`tsx_earn_yld`	0.21002594	0.30465553	0.6894	0.4914156	
$d(\log(\texttt{cpi}), 12)$	−0.14188101	0.35834298	−0.3959	0.6925939	
$d(\texttt{mvx}, 1)$	−0.00208340	0.00059677	−3.4911	0.0005973	***
$d(\log(\texttt{crude\_spot\_cad}), 1)$	−0.06501221	0.03159773	−2.0575	0.0409972	*
$d(\log(\texttt{gold\_cad}), 3)$	0.12403534	0.05284683	2.3471	0.0199457	*
$d(\log(\texttt{silver\_cad}), 12)$	−0.08049292	0.03998297	−2.0132	0.0454995	*
$d(\log(\texttt{gsci\_cad}), 12)$	−0.03600647	0.02741219	−1.3135	0.1905832	
$d(\log(\texttt{leading\_indic}), 1)$	2.36139475	1.20881661	1.9535	0.0522237	.
$d(\texttt{exchange\_rate}, 12)$	0.12878415	0.14177036	0.9084	0.3648118	
$d(\log(\texttt{unemployment}), 12)$	−0.09630010	0.04932094	−1.9525	0.0523380	.
$R^2 : 0.2505$ Adjusted $R^2 : 0.172$	Box-Ljung test results: $\chi^2 = 9.1456, df = 10, p = 0.5183$				



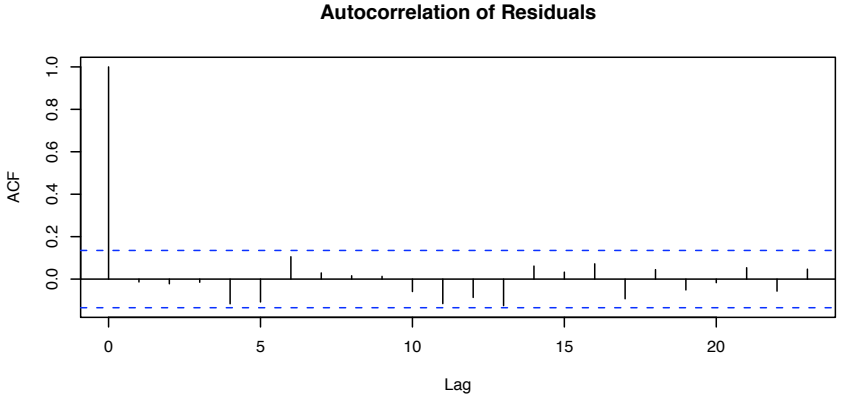
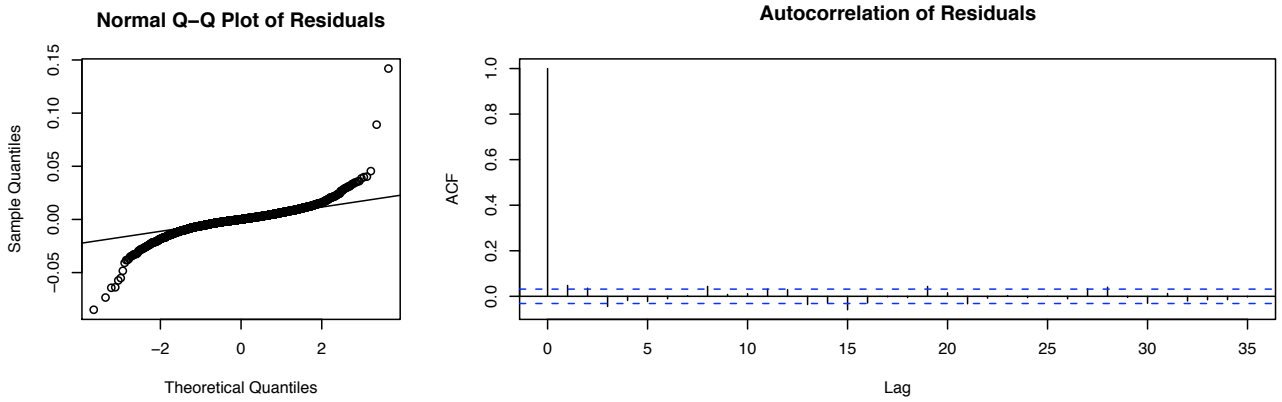


Table 5.24. *Linear regression of daily TSX-Composite returns using Canadian input variables.*

Coefficient	Estimate	Std Err	t Value	$\Pr(>\|t\|)$	
$d(\log(\texttt{tsx\_spot}), 1)$	$-1.3430e-01$	$1.0198e-01$	-1.3170	0.1879160	
$d(\log(\texttt{tsx\_spot}), 5)$	$-2.0794e-02$	$1.6898e-02$	-1.2305	0.2185738	
$d(\log(\texttt{tsx\_spot}), 48 \times 21)$	$-3.2177e-03$	$1.4409e-03$	-2.2331	0.0255969	*
`ca_yield_10yr`	$-3.2156e-04$	$2.3140e-04$	-1.3896	0.1647279	
$d(\texttt{ca\_yield\_10yr} - \texttt{ca\_yield\_3mo}, 504)$	$-3.3356e-04$	$1.3000e-04$	-2.5659	0.0103290	*
$d(\texttt{ca\_yield\_10yr} - \texttt{ca\_yield\_3mo}, 63)$	$-2.4764e-04$	$2.0814e-04$	-1.1898	0.2342167	
`ca_yield_3mo` − `us_yield_3mo`	$-1.2674e-03$	$3.1353e-04$	-4.0425	$5.393e-05$	***
`eurodollar_3mo` − `us_yield_3mo`	$9.8257e-04$	$6.0815e-04$	1.6157	0.1062479	
`tsx_div_yld`	$3.2492e-01$	$9.0952e-02$	3.5724	0.0003580	***
$d(\texttt{tsx\_div\_yld}, 252)$	$-1.0396e-01$	$5.2669e-02$	-1.9739	0.0484636	*
`tsx_earn_yld`	$-2.5382e-02$	$1.5068e-02$	-1.6845	0.0921632	.
$d(\texttt{tsx\_earn\_yld}, 21)$	$1.9000e-02$	$4.9085e-02$	0.3871	0.6987180	
$d(\texttt{mvx}, 1)$	$-5.8948e-04$	$2.5665e-04$	-2.2968	0.0216822	*
$d(\texttt{mvx}, 5)$	$-1.4044e-04$	$1.6259e-04$	-0.8638	0.3877724	
$d(\texttt{mvx}, 21)$	$-5.0766e-05$	$5.0854e-05$	-0.9983	0.3182068	
$d(\texttt{mvx}, 252)$	$-5.9467e-05$	$2.8202e-05$	-2.1086	0.0350455	*
$d(\log(\texttt{crude\_spot\_cad}), 252)$	$1.3335e-03$	$5.9690e-04$	2.2341	0.0255332	*
$d(\log(\texttt{gold\_cad}), 252)$	$4.9162e-03$	$2.1591e-03$	2.2770	0.0228398	*
$d(\log(\texttt{silver\_cad}), 252)$	$-3.6061e-03$	$1.4239e-03$	-2.5326	0.0113629	*
$d(\texttt{exchange\_rate}, 252)$	$7.5513e-03$	$6.2214e-03$	1.2138	0.2249162	
$d(\log(\texttt{unemployment}), 252)$	$-5.5161e-03$	$2.3024e-03$	-2.3958	0.0166317	*
`l_congestion`	$-2.5926e-03$	$1.0635e-03$	-2.4377	0.0148254	*
`l_conseqclose`	$-1.3823e-03$	$4.4334e-04$	-3.1180	0.0018345	**
`l_cup`	$6.6483e-04$	$6.1501e-04$	1.0810	0.2797605	
`l_dom`	$7.9691e-04$	$2.8862e-04$	2.7611	0.0057888	**
`l_midpoint`	$1.3207e-03$	$6.4585e-04$	2.0448	0.0409384	*
`s_cap`	$-2.6160e-03$	$7.0505e-04$	-3.7104	0.0002099	***
R^2 : 0.05508 Adjusted R^2 : 0.04842	Box-Ljung test results: $\chi^2 = 33.1164, df = 10, p = 0.0002604$				



adhered to the following principles to devise the European model: (i) we only considered fundamental variables from the two largest countries of continental Europe (France and Germany); (ii) since the monetary policy of the European Central Bank (for setting interest rates) is closest to that historically followed by the Deutsche Bundesbank, we use the German measures of interest rates an inputs; (iii) all prices are converted to Deutsche Marks, even after the introduction of the Euro where the fixed exchange rate $1\,\text{EUR} = 1.95583\,\text{DEM}$ is used. Furthermore, as noted below, some relatively recent Euro-wide data series are back-extended from country series with the help of regressions.

The following raw variables are used for the European models:

Variable Name	Description
`stoxx_spot`	Dow Jones Eurostoxx 50 Price Index (EUR)[†]
`vstoxx`	Dow Jones Eurostoxx implied volatility index[†]
`de_yield_10yr`	Ten-year German government bond yield
`de_yield_3m`	Three-month German government bond yield
`stoxx_div_yld`	Dow Jones Eurostoxx 50 dividend yield[*]
`crude_spot_dem`	Spot price of crude oil (in Deutsche Marks)
`exchange_rate`	DEM/USD exchange rate
`de_unemployment`	West-Germany total unemployment (seasonally adjusted)[†]
`fr_unemployment`	Metropolitan France total unemployment; excludes DOM/TOM (seasonally adjusted)[‡]

Eurostoxx Monthly Returns

Table 5.27 shows the results of a linear regression of monthly Eurostoxx 50 returns (price index) on a set of inputs derived from the above raw variables. Monthly data over the period April 1986 to November 2004 is used in the regression. The following input variable transformations are provided:

Transformation Notation	Description
$d(\log(\texttt{stoxx\_spot}), 6)$	One-month stocks log-return
$d(\log(\texttt{stoxx\_spot}), 9) - \log(\texttt{vstoxx})$	Log ratio of nine-month stocks return to volatility[§]
$d(\log(\texttt{stoxx\_spot}), 36)$	Three-year stocks log-return
$\texttt{de\_yield\_1yr} - \texttt{de\_yield\_3mo}$	Short-end term spread
$d(\texttt{de\_yield\_10yr} - \texttt{de\_yield\_3m}, 24)$	Two-year increase in long-end term spread
$(\texttt{us\_yield\_3mo} - \texttt{de\_yield\_3mo}) - (\texttt{us\_yield\_10yr} - \texttt{de\_yield\_10yr})$	Difference between three-month and ten-year sovereign spread

[*] See section *"Backcasting" for European Variables* on p. 208 for details on how these variables are computed.

[†] Data source: Bundesbank series `USCY01`.

[‡] Data source: INSEE.

[§] This statistic is predictive of the direction of change, according to Christoffersen and Diebold (2006) and Christoffersen, Diebold, Mariano, Tay, and Tse (2007).

$d(\texttt{stoxx\_div\_yld}, 36)$	Increase in stock dividend yield over the past three years
$d(\texttt{vstoxx}, 1)$	One-month increase in stock market volatility
$d(\log(\texttt{crude\_spot\_dem}), 1)$	One-month increase in (log) crude oil prices
$d(\texttt{exchange\_rate}, 1)$	One-month variation in exchange rate
$d(\texttt{de\_unemployment}, 24)$	Two-year variation in German unemployment
$d(\max(\texttt{fr\_unemployment} - \texttt{de\_unemployment}, 0), 24)$	Two-year floored increase in France versus German unemployment*

Eurostoxx Daily Returns

Table 5.28 shows the results of a linear regression of daily Eurostoxx 50 returns (price index) on inputs similar to those previously described for the monthly-returns model. Daily data over the period 1986-12-15 to 2004-12-31 is used in the regression. Note that all lags are now expressed in days. The autocorrelation and fat tails of residuals effects observed for the U.S. and Canadian models persist for the European model as well.

"Backcasting" for European Variables

Several European variables have been defined in their present form too recently to present the kind of history length required for reliable modeling; while it is desirable to use these variables going forward, a solution must be found to fill in for the missing history. When we have access to other variables highly correlated with the variables of interest, and provided they have sufficient history, a simple regression can be used to retroactively backfill the missing history. For example, suppose that we are interested in using variable X going forward, for which we have a short history, but have access to an alternative variable Y for which a much longer history exist. Further suppose that X and Y are highly contemporaneously correlated, and that this relationship is stable through time. We can start by modeling X as

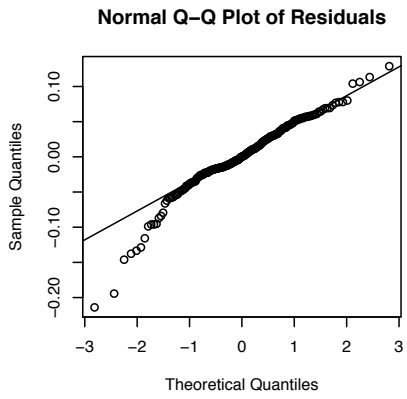
$$X_t = \alpha + \beta Y_t + \epsilon_t$$

over the period for which histories overlap (our short history of X). Then simply use the regression with older values of Y to construct a reasonable estimator of what X would have been. We followed this procedure for three European fundamental variables described next.

*The rationale behind this atypical variable is predicated on the observation that the French government is historically more interventionist than the German one; if the French unemployment rate stays relatively high compared to the German over a relatively long period (two years), government intervention becomes more likely in order to "help the economy", which generally has a positive effect on stocks. Experimentally, we note that the high statistical significance of this variable is very robust to the lag employed and the time period over which we run the regression, but the *direction* of the difference has to be (thresholded) France minus Germany.

Table 5.27. *Linear regression of monthly Eurostoxx returns using European input variables.*

Coefficient	Estimate	Std Err	t Value	$\Pr(>\|t\|)$	
$d(\log(\texttt{stoxx\_spot}), 6)$	−0.04766259	0.03151279	−1.5125	0.1320550	
$d(\log(\texttt{stoxx\_spot}), 9) - \log(\texttt{vstoxx})$	−0.00737028	0.00178717	−4.1240	$5.540e-05$	***
$d(\log(\texttt{stoxx\_spot}), 36)$	−0.02741226	0.01190262	−2.3030	0.0223478	*
$\texttt{de\_yield\_1yr} - \texttt{de\_yield\_3mo}$	−0.04988716	0.01628859	−3.0627	0.0025083	**
$d(\texttt{de\_yield\_10yr} - \texttt{de\_yield\_3m}, 24)$	−0.00426656	0.00265457	−1.6072	0.1096432	
$(\texttt{us\_yield\_3mo} - \texttt{de\_yield\_3mo}) - (\texttt{us\_yield\_10yr} - \texttt{de\_yield\_10yr})$	0.01927209	0.00444584	4.3349	$2.351e-05$	***
$d(\texttt{stoxx\_div\_yld}, 36)$	−1.14786232	0.49194925	−2.3333	0.0206680	*
$d(\texttt{vstoxx}, 1)$	−0.00061588	0.00094682	−0.6505	0.5161637	
$d(\log(\texttt{crude\_spot\_dem}), 1)$	−0.10546979	0.04313407	−2.4452	0.0153806	*
$d(\texttt{exchange\_rate}, 1)$	0.15208525	0.08271935	1.8386	0.0675234	.
$d(\texttt{de\_unemployment}, 24)$	0.02439216	0.00693306	3.5182	0.0005423	***
$d(\max(\texttt{fr\_unemployment} - \texttt{de\_unemployment}, 0), 24)$	0.04135319	0.01130572	3.6577	0.0003286	***
$R^2 : 0.213$ Adjusted $R^2 : 0.1638$	Box-Ljung test results: $\chi^2 = 13.1199, df = 10, p = 0.2170$				



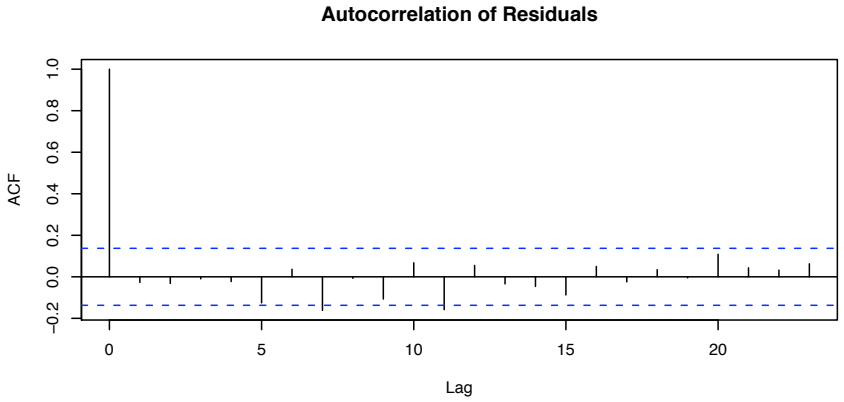
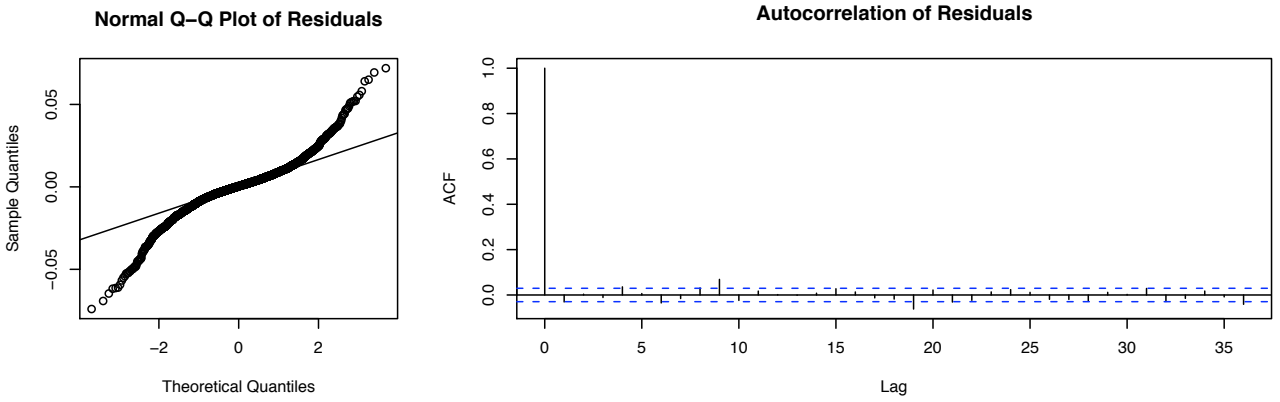


Table 5.28. *Linear regression of daily Eurostoxx returns using European input variables.*

Coefficient	Estimate	Std Err	t Value	$\Pr(>\|t\|)$	
$d(\log(\texttt{stoxx\_spot}), 5)$	$-3.1462e-02$	$1.0261e-02$	-3.0662	0.0021809	**
$d(\log(\texttt{stoxx\_spot}), 9*21)$	$-6.1144e-03$	$2.0671e-03$	-2.9579	0.0031136	**
$d(\log(\texttt{stoxx\_spot}), 36*21)$	$-1.4947e-03$	$7.1455e-04$	-2.0918	0.0365139	*
$d(\texttt{de\_yield\_10yr} - \texttt{de\_yield\_3mo}, 24*21)$	$-6.7734e-04$	$1.7408e-04$	-3.8909	0.0001013	***
$(\texttt{us\_yield\_3mo} - \texttt{de\_yield\_3mo}) - (\texttt{us\_yield\_10yr} - \texttt{de\_yield\_10yr})$	$9.8348e-04$	$1.9355e-04$	5.0813	$3.905e-07$	***
$d(\texttt{stoxx\_div\_yld}, 36*21)$	$-8.8901e-02$	$3.0479e-02$	-2.9168	0.0035543	**
$d(\texttt{vstoxx}, 1)$	$-9.8284e-04$	$2.4869e-04$	-3.9521	$7.868e-05$	***
$d(\texttt{vstoxx}, 21) - d(\texttt{vstoxx}, 12*21)$	$3.0385e-05$	$2.5139e-05$	1.2087	0.2268561	
$d(\log(\texttt{crude\_spot\_dem}), 1*21)$	$-3.7695e-03$	$2.4107e-03$	-1.5637	0.1179611	
$d(\log(\texttt{silver\_spot\_dem}), 1*21)$	$-5.8498e-03$	$3.2279e-03$	-1.8123	0.0700119	.
$d(\log(\texttt{crude\_spot\_dem}), 12*21)$	$-1.3638e-03$	$9.2699e-04$	-1.4712	0.1413056	
$d(\log(\texttt{gold\_spot\_dem}), 12*21)$	$9.2414e-03$	$2.7797e-03$	3.3246	0.0008926	***
$d(\texttt{exchange\_rate}, 1*21)$	$1.7264e-02$	$4.2708e-03$	4.0424	$5.382e-05$	***
$d(\texttt{exchange\_rate}, 12*21)$	$2.1311e-03$	$1.3470e-03$	1.5821	0.1136983	
$d(\texttt{de\_unemployment}, 24*21)$	$2.2257e-03$	$4.3751e-04$	5.0872	$3.785e-07$	***
$d(\max(\texttt{fr\_unemployment} - \texttt{de\_unemployment}, 0), 504)$	$3.3571e-03$	$6.3176e-04$	5.3139	$1.126e-07$	***
`l_congestion`	$9.8983e-04$	$5.0830e-04$	1.9473	0.0515581	.
`l_dom`	$1.1927e-03$	$3.4380e-04$	3.4691	0.0005272	***
`l_month`	$6.7269e-04$	$3.6776e-04$	1.8291	0.0674470	.
`s_cap`	$-4.5482e-04$	$4.7762e-04$	-0.9523	0.3410190	
`s_dow`	$-1.4081e-03$	$4.4856e-04$	-3.1391	0.0017057	**
R^2 : 0.03709 Adjusted R^2 : 0.03249	Box-Ljung test results: $\chi^2 = 44.8, df = 10, p = 2.4e-06$				



▶ **stoxx_spot :** Since the Eurostoxx index only officially starts being defined in 1986-12-31, it is extended backwards to 1980-01-03 by regressing against the DAX (German stock index) and CAC 40 (French stock index, converted to DEM) on the period 1986-12-31 to 2007-12-31. The regression diagnostics read as follows:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	-2.290e+02	6.152e+00	-37.22	<2e-16	***
DAX30	1.985e-01	5.631e-03	35.25	<2e-16	***
CAC40 (DEM)	2.003e+00	2.470e-02	81.09	<2e-16	***
Residual standard error: 175.3 on 5164 degrees of freedom					
Multiple R-squared: 0.9827, Adjusted R-squared: 0.9827					
F-statistic: 1.465e+05 on 2 and 5164 DF, p-value: < 2.2e-16					

▶ **vstoxx :** As previously, the VStoxx only starts in 1999-01-06; over the period 1986-01-01 until 1999-01-05, the log-VStoxx is extended by a linear regression on the log-VIX (CBOE Volatility Index, U.S.). The regression diagnostics are:

	Estimate	Std. Error	t value	Pr(>|t|)
(Intercept)	0.502286	0.023709	21.19	<2e-16 ***
log(VixLow)	0.884464	0.007919	111.68	<2e-16 ***
Residual standard error: 0.1408 on 2368 degrees of freedom				
Multiple R-squared: 0.8404, Adjusted R-squared: 0.8404				
F-statistic: 1.247e+04 on 1 and 2368 DF, p-value: < 2.2e-16				

▶ **stoxx_div_yld :** This variable is defined as the trailing twelve-month dividend yield, computed as the difference between the Eurostoxx Total Return Index and the Price Index. It is only defined as such from January 1989, and is particularly difficult to back-extend; since neither Germany- nor France-specific yield histories were available to the author, we found that a reasonable model is to use a regression on the German 3-month and 1-year bond rates along with the U.S. dividend yield and earnings yield, using data from December 1988 until December 1998 (which constitutes a stable period), yielding:

	Estimate	Std. Error	t value	Pr(>|t|)	
(Intercept)	0.041692	0.001895	22.002	< 2e-16	***
German 3-month yield	0.006579	0.001159	5.678	1.02e-07	***
German 1-year yield	-0.010527	0.001174	-8.969	6.15e-15	***
U.S. Dividend yield	1.002916	0.233241	4.300	3.58e-05	***
U.S. Earnings yield	-0.208445	0.070902	-2.940	0.00396	**
Residual standard error: 0.00421 on 116 degrees of freedom					
Multiple R-squared: 0.5342, Adjusted R-squared: 0.5182					
F-statistic: 33.26 on 4 and 116 DF, p-value: < 2.2e-16					

5.C Appendix: Detailed U.S. Model Results

As outlined in §5.6.2/p. 171, all out-of-sample results presented for the U.S. are split into two disjoint time periods: (i) 1990–1998 (inclusively), and (ii) 1999–2007 (inclusively). The first time period is used for hyperparameter selection, and the second for (indicative) performance evaluation.

The following variables are used in the analyses of variance presented in the next sections. They correspond to the various experimental conditions that were varied:

Table 5.29. *Hyperparameters associated with each model, as studied in the experimental results.*

Variable Name	Model	Description
`Controller.controller.type`	Any	Model Type
`EquityIndexes.input.flavor`	Any	Input Flavor: either `fundamental` or `all` (both fundamental and technical)
`EquityIndexes.rectify.thresh`	Any	Rectification Threshold: value around zero below which a return forecast should yield a neutral position
`EquityIndexes.target.init`	Any	Type of Target Variable (see §5.6.1/p. 169)
`OptClassifNN.nhidden`	Classif. NNet	Nb of Hidden Units
`OptClassifNN.nstages`	Classif. NNet	Nb of Optimization Stages (conjugate gradient optimzation)
`OptClassifNN.weight.decay`	Classif. NNet	Weight Decay
`Classification.nb.classes`	Classif. NNet	Number of Classes
`KBest.K`	Financial NNet	Nb of paths to extract
`OpTrProblem.prop.cost`	Financial NNet	Transaction costs in K-best search
`OptFinancialNNet.hint.penalty`	Financial NNet	Value of λ_{TARG} (*cf.* eq. (5.27)); importance of "target hint" (eq. 5.26) in main training objective
`OptFinancialNNet.nstages`	Financial NNet	Nb of Optimization Stages (conjugate gradient optimzation)
`OptFinancialNNet.prefit.hint.penalty`	Financial NNet	Importance of "target hint" (eq. 5.26) in prefitting training objective
`OptFinancialNNet.prefit.nstages`	Financial NNet	Fraction of Prefitting Stages (from 0 to 1; 0 is no prefitting)
`OptFinancialNNet.prop.cost`	Financial NNet	Transaction costs in final optimization criterion
`OptFinancialNNet.weight.decay`	Financial NNet	Weight Decay, the λ_{WD} hyperparameter in eq. (5.27).
`OptLR.weight.decay`	Linear Regression	Weight Decay
`kernel.options.hyper.ard`	Gaussian Process	Whether kernels have Automatic Relevance Determination
`optimizer.options.nstages`	Gaussian Process	Nb of conjugate-gradient stages in kernel hyperparameter optimization

5.C.1 Linear Regression

Tables 5.30 and 5.31 show the ANOVA tables for the Linear Regression model. We note that both the input type and rectification threshold have a significant and consistent impact on performance. The weight decay is not significant. Looking at the pairwise mean differences (Figs. 5.34 and 5.35) the fundamental set of variables consistently underperforms the set that also includes technical variables. However there is less persistence in the threshold parameter, with 0.03 significantly underperforming in the first period, but among the best in the latter.

The selected hyperparameters on the 1990–1998 period are:

- `EquityIndexes.input.flavor` = `all`
- `EquityIndexes.rectify.thresh` $\in \{0.01, 0.003, 0.001\}$

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	5.5900	1.8633	688.3642	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.0224	0.0224	8.2732	0.004782	**
EquityIndexes.rectify.thresh	3	0.0526	0.0175	6.4829	0.000426	***
OptLR.weight.decay	3	3.988e-09	1.329e-09	4.911e-07	1.000000	
Residuals	117	0.3167	0.0027			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.30. ANOVA *results for United States, Linear Regression model, Period ending in 1998.*

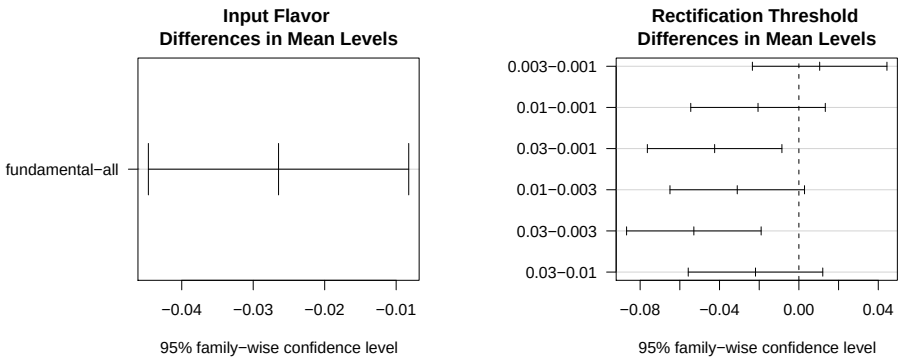
	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	2.95091	0.98364	160.3700	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.08063	0.08063	13.1454	0.0004281	***
EquityIndexes.rectify.thresh	3	0.12889	0.04296	7.0047	0.0002252	***
OptLR.weight.decay	3	4.471e-06	1.490e-06	0.0002	0.9999947	
Residuals	117	0.71763	0.00613			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

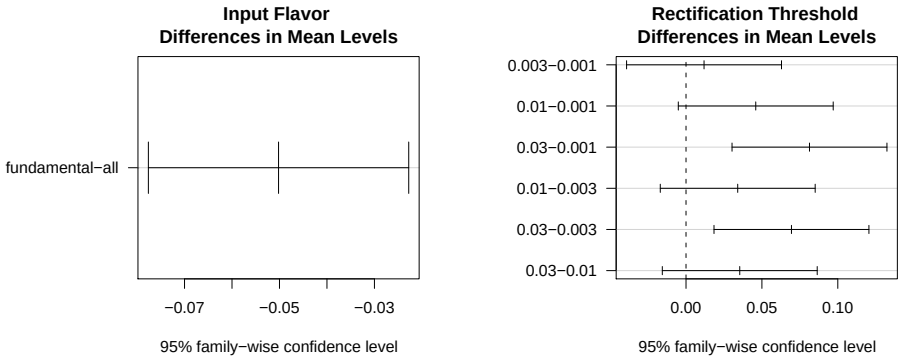
Table 5.31. ANOVA *results for United States, Linear Regression model, Period ending in 2007.*

5.C.2 Gaussian Process

Tables 5.32 and 5.33 show the ANOVA tables for the Gaussian Process model. The type of target variable has a very significant and persistent impact on performance, whereas the impact of other hyperparameters is more mixed. Looking at the pairwise mean differences (Figs. 5.36 and 5.37), using the set of all variables appears preferable to fundamental alone (although not significantly in the second period). The K-best targets are consistently significantly *worse* than `RETURN-DAILY`, `RETURN-MONTHLY` or `SIGN-DAILY` targets.



▶ **Figure 5.34.** *Hyperparameter comparison for Linear Regression models. United States, 1990–1998.*



▶ **Figure 5.35.** *Hyperparameter comparison for Linear Regression models. United States, 1999–2007.*

The impact of the number of optimization stages suggests that only one iteration of conjugate gradient is not sufficient (in the first period), although there are no significant difference otherwise.

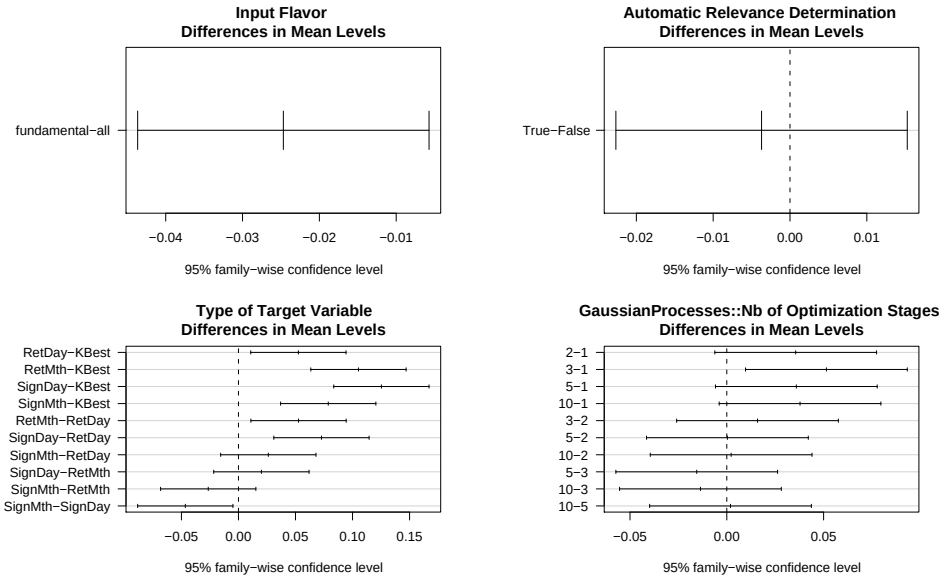
The selected hyperparameters on the 1990–1998 period are:

- `EquityIndexes.input.flavor` = `all`
- `optimizer.options.nstages` $\neq 1$
- `EquityIndexes.target.init` $\in$ {`RETURN-MONTHLY`, `SIGN-DAILY`}

Table 5.32. ANOVA *results for United States, Gaussian Process model, Period ending in 1998.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	1.7068	0.5689	30.4820	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.1219	0.1219	6.5312	0.01079	*
EquityIndexes.rectify.thresh	1	0.0002	0.0002	0.0097	0.92144	
EquityIndexes.target.init	4	1.5280	0.3820	20.4665	4.685e-16	***
kernel.options.hyper.ard	1	0.0027	0.0027	0.1467	0.70179	
optimizer.options.nstages	4	0.2342	0.0585	3.1369	0.01421	*
Residuals	785	14.6516	0.0187			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1



◀ **Figure 5.36.** *Hyperparameter comparison for Gaussian Process models. United States, 1990–1998.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	9.8733	3.2911	275.0647	<2e-16	***
EquityIndexes.input.flavor	1	0.0212	0.0212	1.7691	0.1839	
EquityIndexes.rectify.thresh	1	4.47e-06	4.47e-06	0.0004	0.9846	
EquityIndexes.target.init	4	1.6436	0.4109	34.3432	<2e-16	***
kernel.options.hyper.ard	1	0.0560	0.0560	4.6809	0.0308	*
optimizer.options.nstages	4	0.0133	0.0033	0.2782	0.8921	
Residuals	785	9.3924	0.0120			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

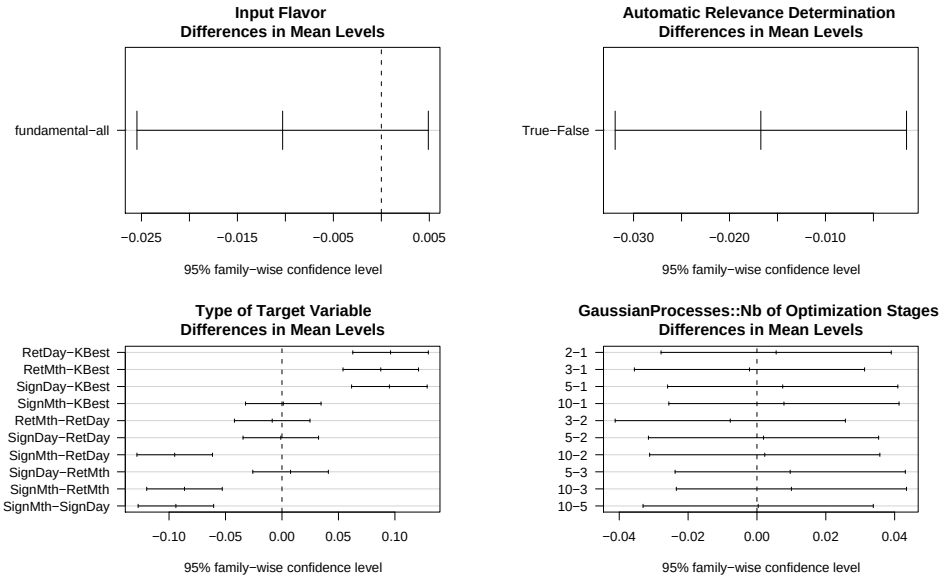
Table 5.33. ANOVA *results for United States, Gaussian Process model, Period ending in 2007.*

5.C.3 Classification Neural Network

Tables 5.34 and 5.35 show the ANOVA tables for the Classification Neural Network model. The target type and number of hidden units have a persistent significant effect, and the number of classes "nearly" so. Looking at the pairwise mean differences (Figs. 5.38 and 5.39), we note that either `SIGN-DAILY` or `SIGN-MONTHLY` consistently beat using the K-best targets, and that no hidden units (in effect, resulting in a simple linear model with a softmax output layer) performs consistently better than higher-capacity models. The three-class problem formulation (long/neutral/short) performs better than the two-class one, although the difference is not significant (by little) in the second period.

The selected hyperparameters on the 1990–1998 period are:

- `Classification.nb.classes` = 3



▶ **Figure 5.37.** *Hyperparameter comparison for Gaussian Process models. United States, 1999–2007.*

- `EquityIndexes.target.init` $\neq$ `KBest`
- `OptClassifNN.weight.decay` $= 10^{-2}$
- `OptClassifNN.nhidden` $= 0$

Table 5.34. ANOVA *results for United States, Classification Neural Network model, Period ending in 1998.*

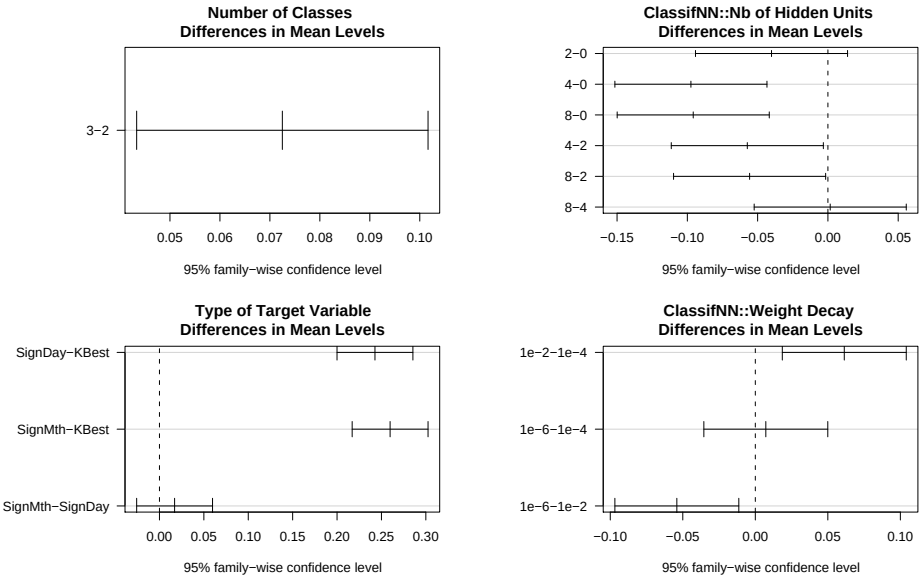
	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	8.6748	2.8916	91.1215	< 2.2e-16	***
Classification.nb.classes	1	0.7568	0.7568	23.8486	1.360e-06	***
EquityIndexes.input.flavor	1	0.0259	0.0259	0.8154	0.366913	
EquityIndexes.target.init	2	8.1064	4.0532	127.7268	< 2.2e-16	***
OptClassifNN.nhidden	3	0.9605	0.3202	10.0896	1.750e-06	***
OptClassifNN.weight.decay	2	0.4314	0.2157	6.7975	0.001210	**
Residuals	563	17.8659	0.0317			

Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Table 5.35. ANOVA *results for United States, Classification Neural Network model, Period ending in 2007.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	3.2129	1.0710	37.1647	< 2.2e-16	***
Classification.nb.classes	1	0.0735	0.0735	2.5490	0.1109	
EquityIndexes.input.flavor	1	0.0482	0.0482	1.6720	0.1965	
EquityIndexes.target.init	2	2.7331	1.3665	47.4222	< 2.2e-16	***
OptClassifNN.nhidden	3	0.8036	0.2679	9.2961	5.234e-06	***
OptClassifNN.weight.decay	2	0.0035	0.0017	0.0600	0.9418	
Residuals	563	16.2238	0.0288			

Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1



◀ **Figure 5.38.** *Hyperparameter comparison for Classification Neural Networks models. United States, 1990–1998.*

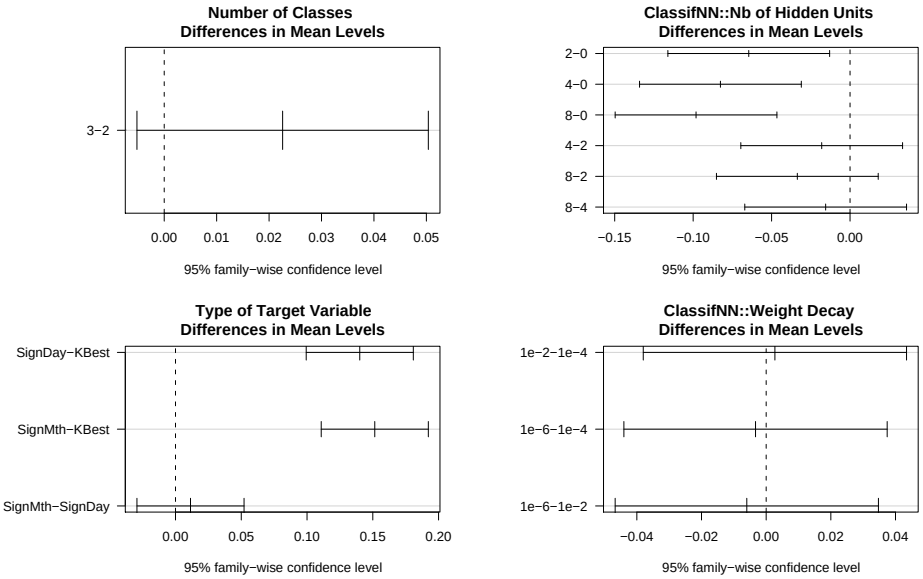
5.C.4 Financial Neural Network

Since higher-order interactions between most hyperparameters and target type are quite complex, we separate out the analysis between two separate settings: (i) using K-Best targets, and (ii) using monthly and daily return-sign targets. Furthermore, a number of preliminary experiments (not reported here) indicate that incorporating hidden units systematically degrades performance out-of-sample, regardless of the number included (from two up to ten) and the amount of regularization applied. Hence, all results reported here are for a FinancialNNet with *zero hidden units*, which consists in nothing more than a core linear model with adjustable output thresholds, trained on a financial criterion.

Tables 5.36 and 5.37 show the ANOVA tables for the FinancialNNet model **using K-best targets** (which are used as part of the "hint term", eq. (5.26)), whereas tables 5.38 and 5.39 show them for `SIGN-DAILY` and `SIGN-MONTHLY` targets.

Considering K-best targets alone, the selected hyperparameters on the 1990–1998 period are (the pairwise mean difference plots are omitted for brevity):

- `EquityIndexes.target.init` = `KBest`
- `OptFinancialNNet.prefit.nstages` $\neq 0.0$
- `OptFinancialNNet.prop.cost` $= 0.002$
- `OptFinancialNNet.weight.decay` $= 10^{-4}$
- `EquityIndexes.input.flavor` = `all`



▶ **Figure 5.39.** *Hyperparameter comparison for Classification Neural Networks models. United States, 1999–2007.*

- `OptFinancialNNet.hint.penalty` $= 0.001$
- `OpTrProblem.prop.cost` $\notin \{0.025, 0.03\}$

Considering the *monthly and daily returns* targets, the selected hyperparameters on the 1990–1998 period are:

- `EquityIndexes.input.flavor` = `all`
- `EquityIndexes.target.init` = `SignDay`
- `OptFinancialNNet.prop.cost` $= 0.002$

A direct comparison between the above two set of hyperparameters yields a non-significant mean Sharpe ratio difference of 0.0053 ($p = 0.60$) in favor of K-best targets over `SignDay` targets for the 1990–1998 period. However, this difference is reversed over the 1999–2007 period, where we obtain a significant mean difference of 0.02 ($p = 0.012$) in favor of `SignDay`. Nevertheless, since the choice of hyperparameters is carried out on the period ending in 1998, we use the union of the above two hyperparameters in the final model comparison to be carried out below.

The pairwise mean differences for the resulting union appear in Figs. 5.40 and 5.41.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	24.8575	8.2858	2466.4422	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.3061	0.3061	91.1286	< 2.2e-16	***
KBest.K	1	0.0023	0.0023	0.6805	0.409672	
OpTrProblem.prop.cost	5	0.0574	0.0115	3.4167	0.004642	**
OptFinancialNNet.hint.penalty	1	0.0874	0.0874	26.0256	4.267e-07	***
OptFinancialNNet.nstages	1	0.0016	0.0016	0.4689	0.493722	
OptFinancialNNet.prefit.nstages	1	0.0007	0.0007	0.2230	0.636933	
Residuals	754	2.5330	0.0034			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.36. ANOVA *results for United States, Financial Neural Network (with K-best targets), Period ending in 1998, after isolating all interactions.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	33.858	11.286	6311.7219	<2e-16	***
EquityIndexes.input.flavor	1	0.002	0.002	1.1816	0.2774	
KBest.K	1	0.001	0.001	0.5841	0.4450	
OpTrProblem.prop.cost	5	0.008	0.002	0.8767	0.4962	
OptFinancialNNet.hint.penalty	1	0.224	0.224	125.4660	<2e-16	***
OptFinancialNNet.nstages	1	0.0003968	0.0003968	0.2219	0.6377	
OptFinancialNNet.prefit.nstages	1	0.001	0.001	0.3213	0.5710	
Residuals	754	1.348	0.002			

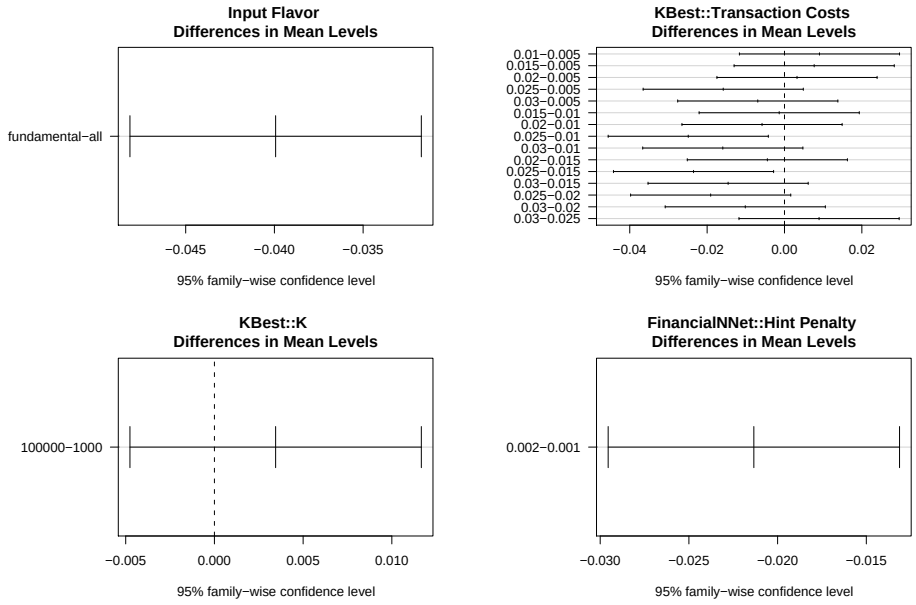
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.37. ANOVA *results for United States, Financial Neural Network (with K-best targets), Period ending in 2007, after isolating all interactions.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	16.8927	5.6309	1007.3539	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.0160	0.0160	2.8686	0.09088	.
EquityIndexes.target.init	1	0.0190	0.0190	3.3980	0.06580	.
OptFinancialNNet.hint.penalty	1	0.0016	0.0016	0.2821	0.59551	
OptFinancialNNet.nstages	1	1.151e-06	1.151e-06	0.0002	0.98856	
OptFinancialNNet.prefit.nstages	2	0.0008	0.0004	0.0684	0.93386	
OptFinancialNNet.prop.cost	2	0.3401	0.1701	30.4237	2.834e-13	***
Residuals	564	3.1526	0.0056			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.38. ANOVA *results for United States, Financial Neural Network (without K-best targets), Period ending in 1998.*



▶ **Figure 5.40.** *Hyperparameter comparison for Financial Neural Networks models. United States, 1990–1998.*

Table 5.39. ANOVA *results for United States, Financial Neural Network (without K-best targets), Period ending in 2007.*

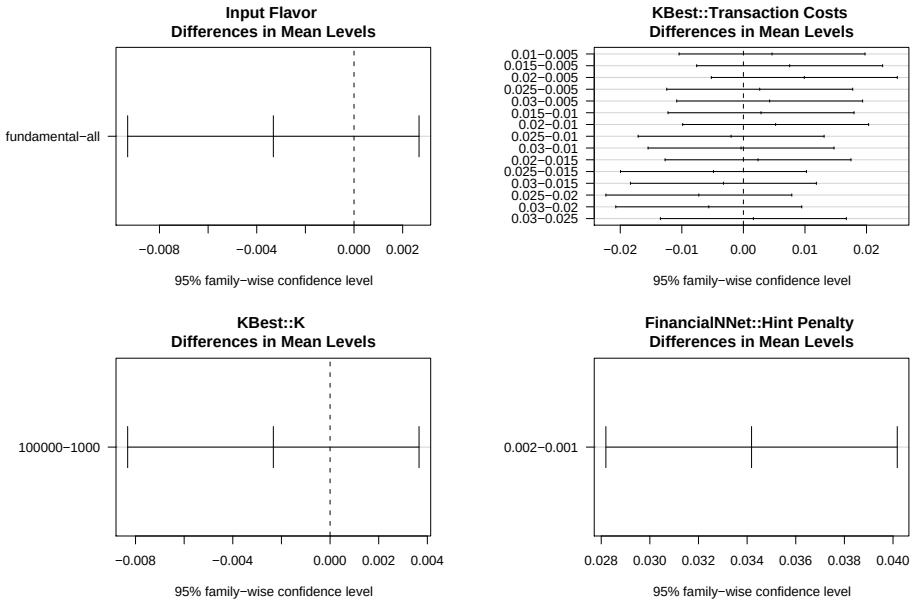
	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	12.6002	4.2001	602.0873	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.1671	0.1671	23.9592	1.286e-06	***
EquityIndexes.target.init	1	0.2446	0.2446	35.0687	5.535e-09	***
OptFinancialNNet.hint.penalty	1	0.0710	0.0710	10.1738	0.001504	**
OptFinancialNNet.nstages	1	0.0005	0.0005	0.0757	0.783285	
OptFinancialNNet.prefit.nstages	2	0.0003	0.0002	0.0223	0.977946	
OptFinancialNNet.prop.cost	2	0.1786	0.0893	12.8030	3.650e-06	***
Residuals	564	3.9344	0.0070			

Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

5.C.5 Overall Results

To obtain an overall comparison between models, we consider, for each model, the set of hyperparameters that were deemed “not significantly different” in the previous sections. We then measure the performance distribution of those models, in two different ways:

- The Sharpe ratio, computed from monthly returns over the complete periods 1990–1998 and 1999–2007.
- Same as above, but where each period is split into four disjoint subperiods of an equal number of months. The Sharpe ratio is separately computed for each subperiod.

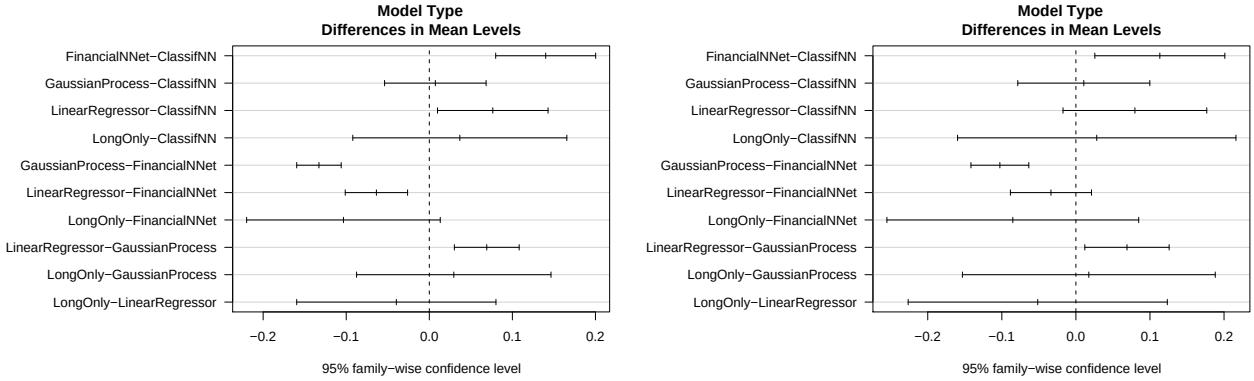


◀ **Figure 5.41.** *Hyperparameter comparison for Financial Neural Networks models. United States, 1999–2007.*

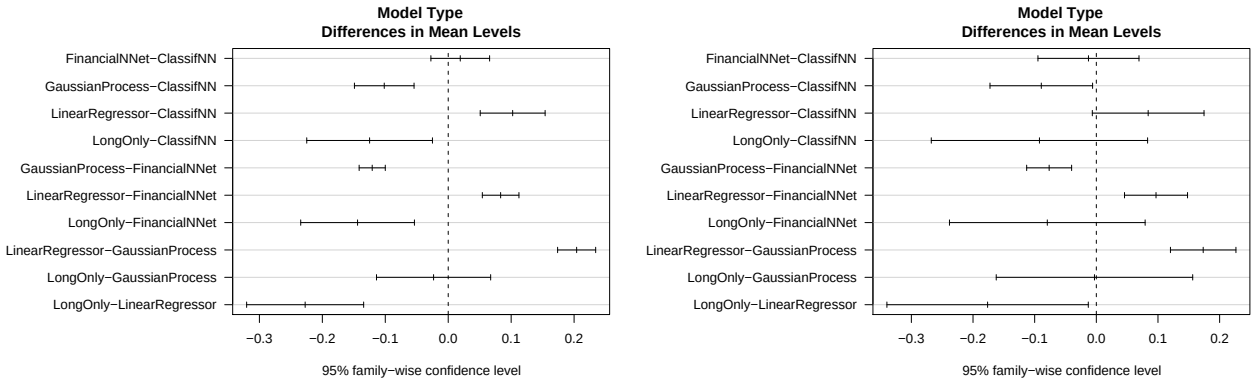
The latter computation serves to confirm the robustness of the former results.

Figures 5.42 and 5.43 illustrate the pairwise mean differences between all models, respectively over the 1990–1998 and 1999–2007 periods. In general, we note little difference between the average performance of the "one-split" and "four-split" results, suggesting a certain consistency in the average Sharpe ratio. (The increase in variance, reflected in the wider error bars, is to be expected).

The most distinct feature of those results is that the best-performing model over 1990–1998, FinancialNNet, switches place with Linear Regression over the 1999–2007 horizon. Recall that the FinancialNNet models all have zero hidden units and have, approximately, the same capacity as the linear regressor, the latter being perhaps more highly regularized due to its robust training criterion and fixed output thresholds. The results illustrate the importance of aggressive regularization to obtaining good out-of-sample performance.



▲ **Figure 5.42.** *United States, 1990–1998.* ***Left:*** *Average pairwise Sharpe ratio difference between all models, single split.* ***Right:*** *Same, but where the evaluation horizon is split into four disjoint subperiods as described in the text.*



▲ **Figure 5.43.** *United States, 1999–2007.* ***Left:*** *Average pairwise Sharpe ratio difference between all models, single split.* ***Right:*** *Same, but where the evaluation horizon is split into four disjoint subperiods as described in the text.*

5.D Appendix: Detailed Canadian Model Results

Results for the Canadian model follow the same pattern as the United States presented in §5.C/p. 212. In particular, we follow the same methodology of splitting the test into two disjoint periods (1991–1998 and 1999–2007), using the first for hyperparameter selection and the second for (indicative) performance evaluation.

The variable names used in the analyses of variance are described on p. 212.

5.D.1 Linear Regression

Tables 5.40 and 5.41 show the ANOVA tables for the Linear Regression model. We note that the target type has a significant and consistent impact on performance, whereas the impact of the type of input variables and rectification threshold is not consistent. The weight decay is not significant. Looking at the pairwise mean differences (Figs. 5.44 and 5.45), the use of monthly returns instead of daily ones proves to be, by a large and significant margin, the best choice. The impact of other hyperparameters is rather unremarkable.

The selected hyperparameters on the 1991–1998 period are:

- `EquityIndexes.rectify.thresh` = 0.01
- `EquityIndexes.target.init` = `RetMth`

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	13.4128	4.4709	584.8103	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.0002	0.0002	0.0301	0.8625	
EquityIndexes.rectify.thresh	6	0.2515	0.0419	5.4831	1.847e-05	***
EquityIndexes.target.init	1	0.6376	0.6376	83.4013	< 2.2e-16	***
OptLR.weight.decay	3	2.020e-06	6.732e-07	0.0001	1.0000	
Residuals	381	2.9128	0.0076			

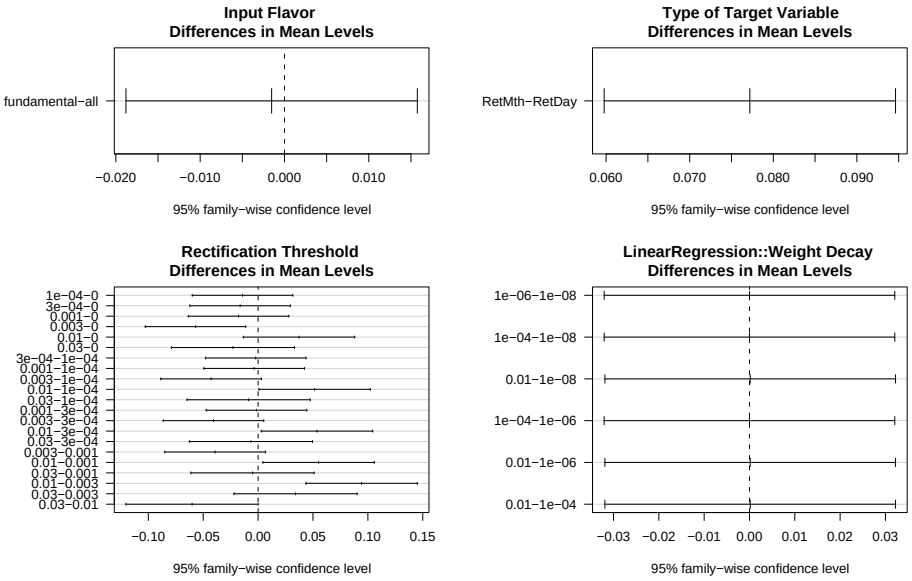
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.40. ANOVA *results for Canada, Linear Regression model, Period ending in 1998.*

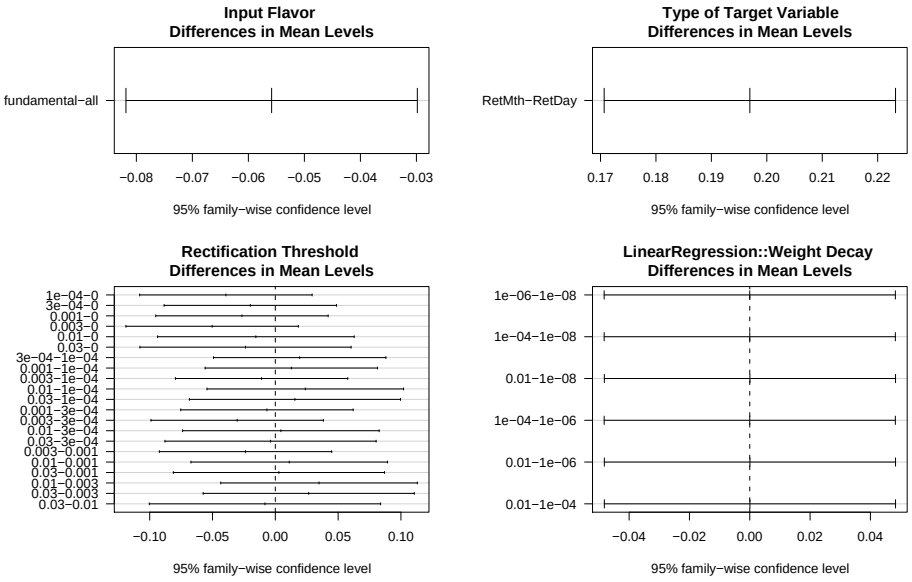
	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	18.3502	6.1167	356.6883	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.3056	0.3056	17.8231	3.042e-05	***
EquityIndexes.rectify.thresh	6	0.0994	0.0166	0.9661	0.448	
EquityIndexes.target.init	1	4.1517	4.1517	242.1027	< 2.2e-16	***
OptLR.weight.decay	3	2.487e-07	8.291e-08	4.835e-06	1.000	
Residuals	377	6.4651	0.0171			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.41. ANOVA *results for Canada, Linear Regression model, Period ending in 2007.*



▶ **Figure 5.44.** *Hyperparameter comparison for Linear Regression models. Canada, 1991–1998.*



▶ **Figure 5.45.** *Hyperparameter comparison for Linear Regression models. Canada, 1999–2007.*

5.D.2 Gaussian Process

Tables 5.42 and 5.43 show the ANOVA tables for the Gaussian Process model. Only the type of target variable shows a significant and consistent impact. Looking at the pairwise mean differences (Figs. 5.46 and 5.47), we note that over the 1991–1998 period, the `SIGN-DAILY` target type is clearly dominant, and significant by a large margin. However, this does not persist so outstandingly in the 1999-2007 period, where the `RETURN-MONTHLY` target is better (although not significantly so against all alternatives).

The selected hyperparameters on the 1991–1998 period are:

- `EquityIndexes.target.init` = `SignDay`

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	0.6025	0.2008	7.4489	9.729e-05	***
EquityIndexes.input.flavor	1	0.0017	0.0017	0.0633	0.8016	
EquityIndexes.target.init	4	0.9483	0.2371	8.7931	1.586e-06	***
optimizer.options.nstages	4	0.0453	0.0113	0.4200	0.7941	
Residuals	187	5.0418	0.0270			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.42. ANOVA *results for Canada, Gaussian Process model, Period ending in 1998.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	8.2695	2.7565	262.0494	<2e-16	***
EquityIndexes.input.flavor	1	0.0003	0.0003	0.0275	0.8686	
EquityIndexes.target.init	4	1.0729	0.2682	25.4986	<2e-16	***
optimizer.options.nstages	4	0.0364	0.0091	0.8645	0.4863	
Residuals	187	1.9670	0.0105			

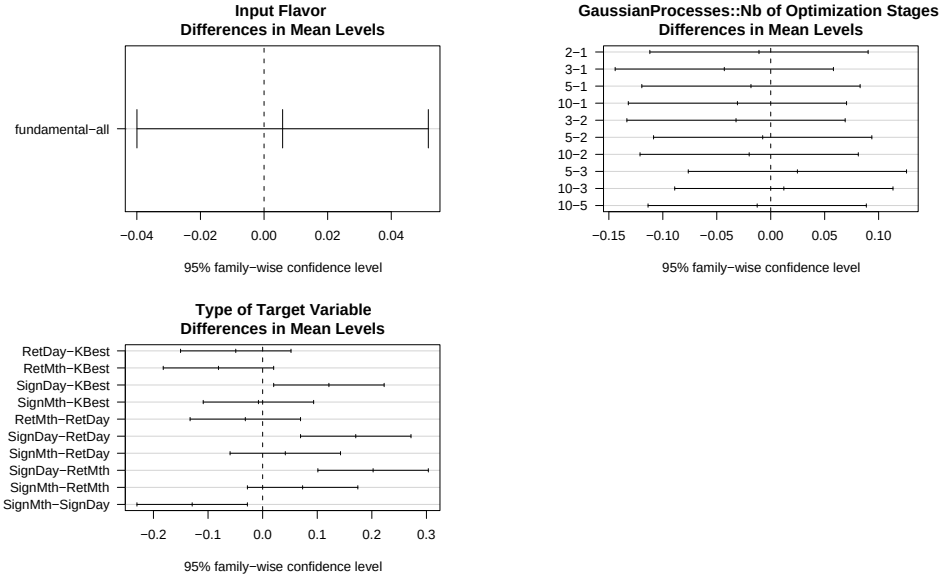
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.43. ANOVA *results for Canada, Gaussian Process model, Period ending in 2007.*

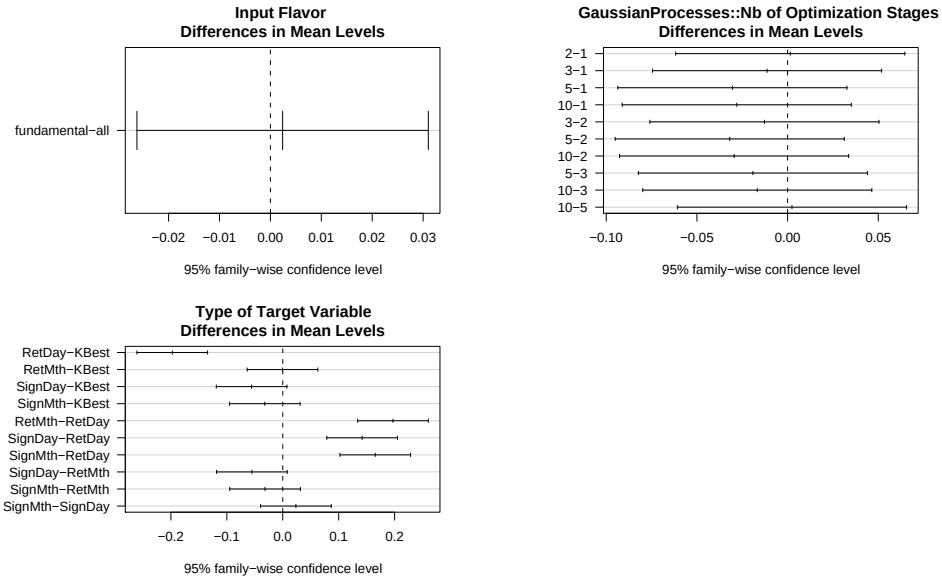
5.D.3 Classification Neural Network

Tables 5.44 and 5.45 show the ANOVA tables for the Classification Neural Network model. The type of target variable and the number units have a very significant and persistent impact; the number of problem classes (two versus three) shows "near" significance. Looking at the pairwise mean differences (Figs. 5.48 and 5.49), the preferred target type is `SIGN-DAILY` in the first time period, whereas `SIGN-MONTHLY` performs better during the second. Both consistently beat the `PURE-KBEST` targets. Regarding the impact of hidden units, having none (i.e. a linear classification model) beats any increased capacity.

The selected hyperparameters on the 1991–1998 period are:



► **Figure 5.46.** *Hyperparameter comparison for Gaussian Process models. Canada, 1991–1998.*



► **Figure 5.47.** *Hyperparameter comparison for Gaussian Process models. Canada, 1999–2007.*

- `Classification.nb.classes` $= 3$
- `EquityIndexes.target.init` $\neq$ `KBest`
- `OptClassifNN.nhidden` $= 0$

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	1.0804	0.3601	8.9871	7.466e-06	***
Classification.nb.classes	1	0.0884	0.0884	2.2052	0.137963	
EquityIndexes.input.flavor	1	0.0044	0.0044	0.1107	0.739432	
EquityIndexes.target.init	2	7.6785	3.8392	95.8098	< 2.2e-16	***
OptClassifNN.nhidden	3	0.6533	0.2178	5.4341	0.001062	**
OptClassifNN.weight.decay	3	0.1676	0.0559	1.3939	0.243401	
Residuals	754	30.2138	0.0401			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.44. ANOVA *results for Canada, Classification Neural Network model, Period ending in 1998.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	22.3957	7.4652	222.0690	< 2.2e-16	***
Classification.nb.classes	1	0.0860	0.0860	2.5593	0.1101	
EquityIndexes.input.flavor	1	0.0014	0.0014	0.0416	0.8385	
EquityIndexes.target.init	2	3.6199	1.8100	53.8409	< 2.2e-16	***
OptClassifNN.nhidden	3	0.7215	0.2405	7.1537	9.681e-05	***
OptClassifNN.weight.decay	3	1.7219	0.5740	17.0736	9.774e-11	***
Residuals	754	25.3470	0.0336			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

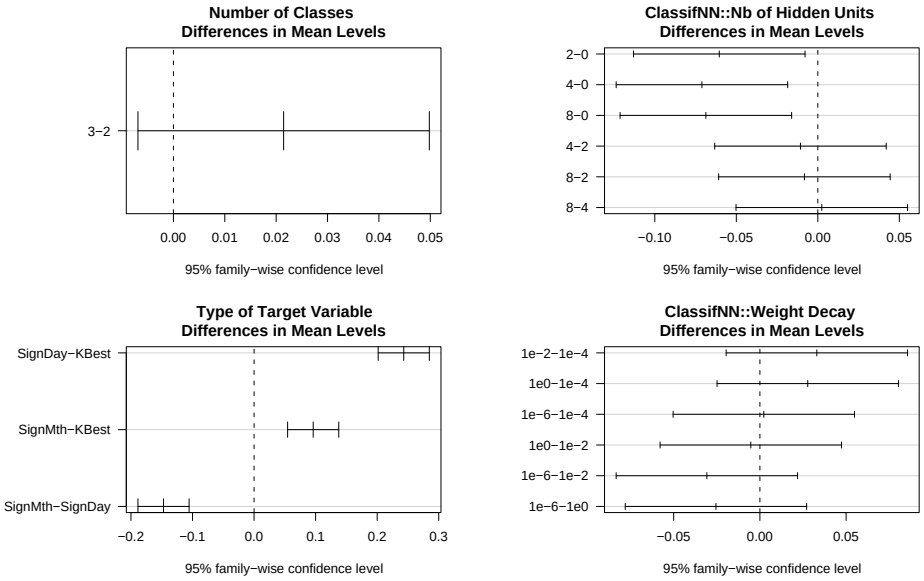
Table 5.45. ANOVA *results for Canada, Classification Neural Network model, Period ending in 2007.*

5.D.4 Financial Neural Network

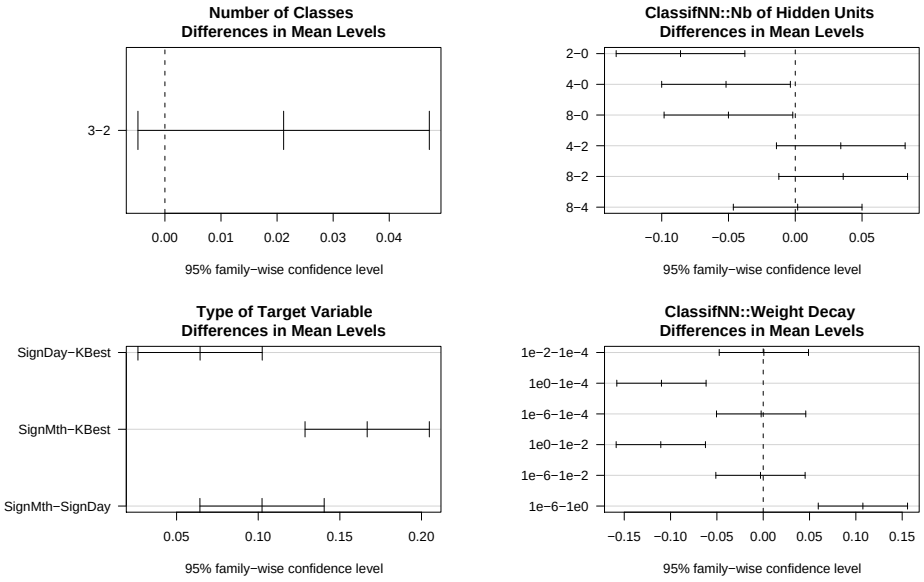
As for the U.S. results, we separate out the analysis between two separate settings: (i) using K-Best targets, and (ii) using monthly and daily return-sign targets. Tables 5.46 and 5.47 show the ANOVA tables for the FinancialNNet model **using K-best targets**, whereas Tables 5.48 and 5.49 show them for `SIGN-DAILY` and `SIGN-MONTHLY` targets.

Considering K-best targets alone, the selected hyperparameters on the 1991–1998 period are (the pairwise mean difference plots are omitted for brevity):

- `EquityIndexes.target.init` $=$ `KBest`
- `OpTrProblem.prop.cost` $\in \{0.005, 0.01\}$
- `OptFinancialNNet.prefit.nstages` $\neq 0.0$
- `OptFinancialNNet.prop.cost` $= 0$



► **Figure 5.48.** *Hyper-parameter comparison for Classification Neural Networks models. Canada, 1991–1998.*



► **Figure 5.49.** *Hyper-parameter comparison for Classification Neural Networks models. Canada, 1999–2007.*

- `OptFinancialNNet.weight.decay` $\neq 10^{-2}$

Considering the *monthly and daily returns* targets, the selected hyperparameters on the 1991–1998 period are:

- `EquityIndexes.target.init` $\neq$ `KBest`
- `OptFinancialNNet.prefit.nstages` $\neq 0.0$
- `OptFinancialNNet.prop.cost` $= 0$
- `OptFinancialNNet.weight.decay` $\neq 10^{-2}$

A direct comparison between these two subsets (looking only at differences in the effect of `EquityIndexes.target.init`) indicates that there are no significant difference between the type of target, on neither period.

The pairwise mean differences for the resulting union appear in Figures 5.50 and 5.51. We note the importance of the prefitting stage: performance degrades significantly (in both periods) if no prefitting is performed. Moreover, excessive weight decay is seen to consistently impact results. The effect of transaction costs in the financial criterion is not so clear: over the 1991–98 period, non-zero costs degrade performance, whereas over the 1999–2007 period, small costs turn out to be (marginally significantly) beneficial.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	17.046	5.682	115.1287	< 2e-16	***
EquityIndexes.input.flavor	1	0.005	0.005	0.0920	0.76167	
OpTrProblem.prop.cost	3	0.552	0.184	3.7255	0.01092	*
OptFinancialNNet.hint.penalty	2	0.103	0.051	1.0390	0.35395	
OptFinancialNNet.prefit.nstages	2	16.430	8.215	166.4572	< 2e-16	***
OptFinancialNNet.prop.cost	2	14.307	7.153	144.9453	< 2e-16	***
OptFinancialNNet.weight.decay	2	4.156	2.078	42.1066	< 2e-16	***
Residuals	2576	127.132	0.049			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.46. ANOVA *results for Canada, Financial Neural Network model (with K-best targets), Period ending in 1998.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	46.820	15.607	263.0151	< 2.2e-16	***
EquityIndexes.input.flavor	1	1.190	1.190	20.0586	7.838e-06	***
OpTrProblem.prop.cost	3	0.031	0.010	0.1716	0.9156	
OptFinancialNNet.hint.penalty	2	0.041	0.021	0.3495	0.7051	
OptFinancialNNet.prefit.nstages	2	37.228	18.614	313.6964	< 2.2e-16	***
OptFinancialNNet.prop.cost	2	22.417	11.208	188.8905	< 2.2e-16	***
OptFinancialNNet.weight.decay	2	12.892	6.446	108.6327	< 2.2e-16	***
Residuals	2576	152.854	0.059			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.47. ANOVA *results for Canada, Financial Neural Network model (with K-best targets), Period ending in 2007.*

Table 5.48. ANOVA *results for Canada, Financial Neural Network model (without K-best targets), Period ending in 1998.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	6.713	2.238	47.0923	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.002	0.002	0.0457	0.8308	
EquityIndexes.target.init	1	0.051	0.051	1.0804	0.2988	
OptFinancialNNet.hint.penalty	2	0.161	0.081	1.6974	0.1836	
OptFinancialNNet.prefit.nstages	2	7.583	3.791	79.7918	< 2.2e-16	***
OptFinancialNNet.prop.cost	2	6.518	3.259	68.5896	< 2.2e-16	***
OptFinancialNNet.weight.decay	2	2.081	1.041	21.9003	4.443e-10	***
Residuals	1282	60.914	0.048			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

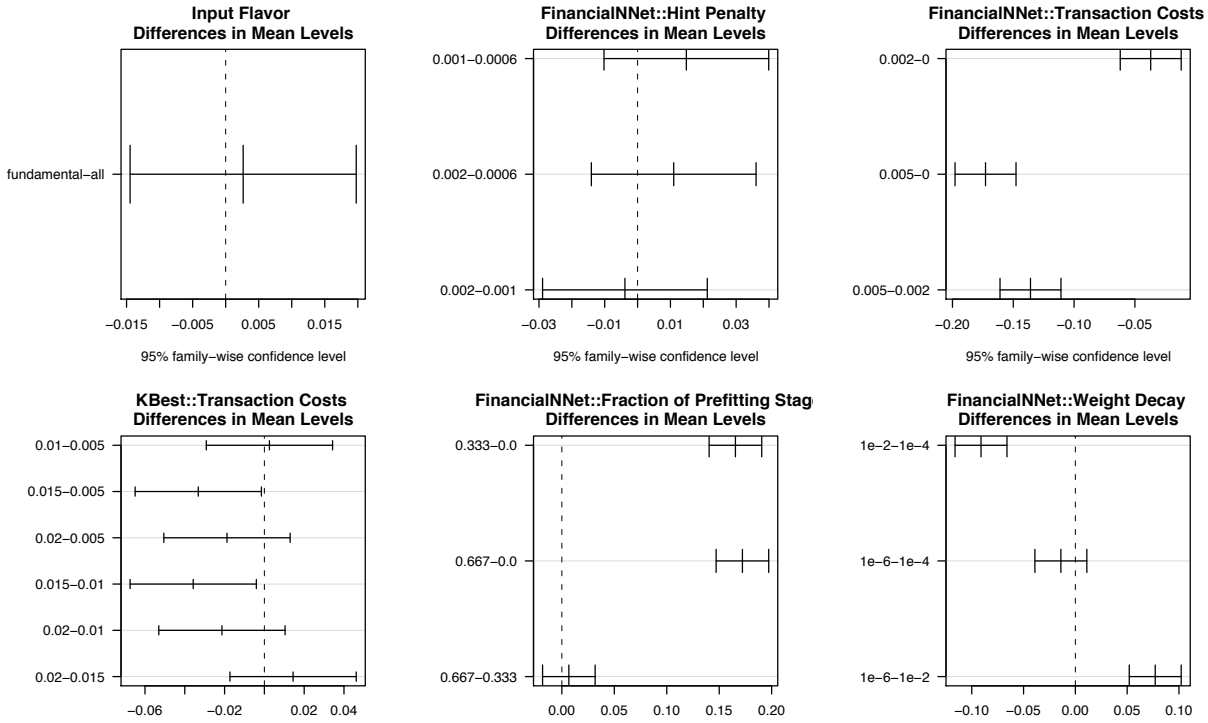
Table 5.49. ANOVA *results for Canada, Financial Neural Network model (without K-best targets), Period ending in 2007.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	24.304	8.101	132.8675	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.507	0.507	8.3208	0.003985	**
EquityIndexes.target.init	1	0.026	0.026	0.4245	0.514807	
OptFinancialNNet.hint.penalty	2	0.002	0.001	0.0170	0.983150	
OptFinancialNNet.prefit.nstages	2	20.586	10.293	168.8114	< 2.2e-16	***
OptFinancialNNet.prop.cost	2	11.855	5.928	97.2188	< 2.2e-16	***
OptFinancialNNet.weight.decay	2	5.052	2.526	41.4312	< 2.2e-16	***
Residuals	1282	78.166	0.061			

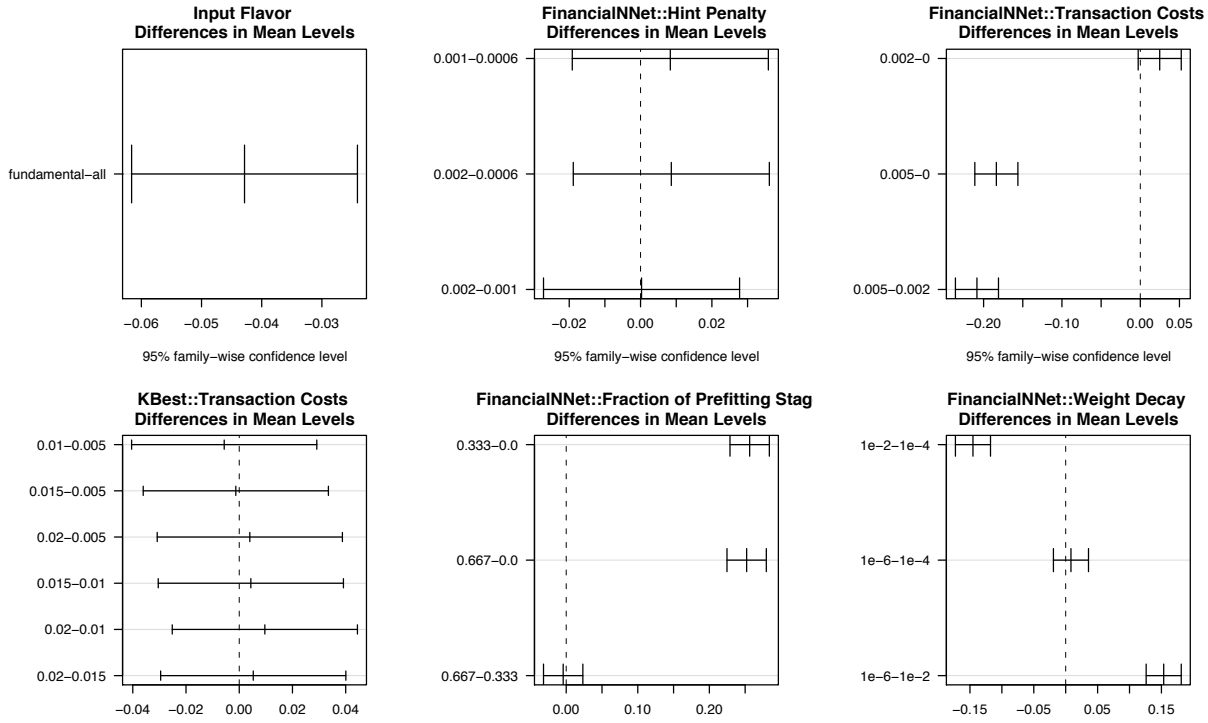
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

5.D.5 Overall Results

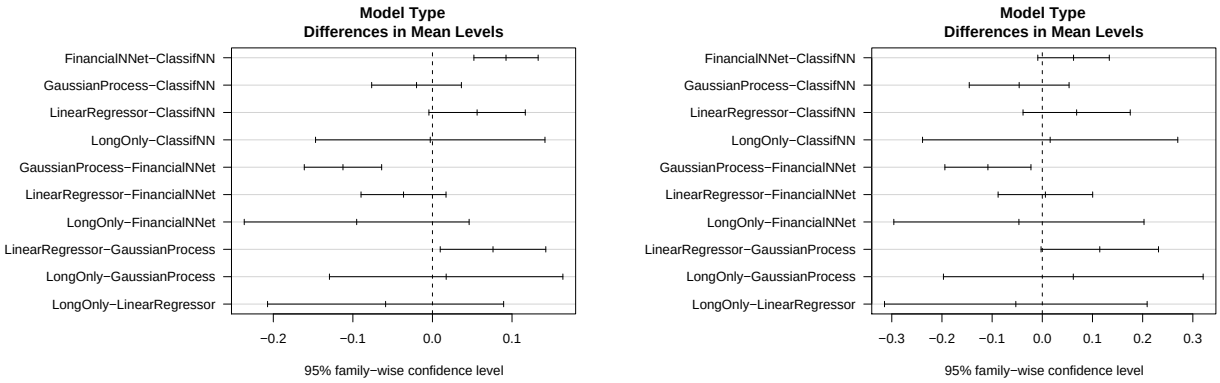
We follow the same procedure as for the U.S. to compare between models, using for each the selected set of hyperparameters. Figures 5.52 and 5.53 illustrate the pairwise mean differences between all models, respectively over the 1991–1998 and 1999–2007 periods. Both the "one-split" and "four-split" results suggest the same relative rankings, during 1991–1998, with the FinancialNNet and Linear Regressor being roughly equivalent to the buy-and-hold model and Classification neural network and Gaussian process models exhibiting worse performance. During the 1999–2007 period both the Linear Regressor and ClassificationNNet perform best.



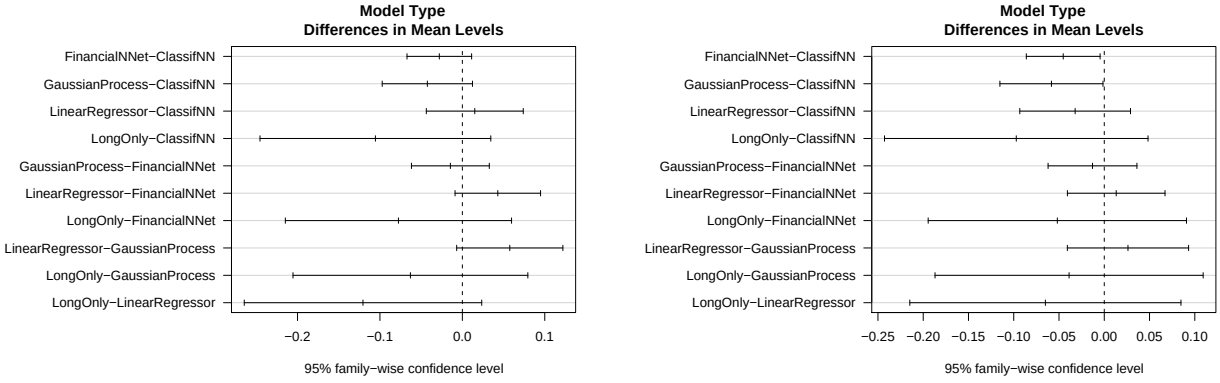
▲ **Figure 5.50.** *Hyperparameter comparison for Financial Neural Networks models. Canada, 1991–1998.*



▲ **Figure 5.51.** *Hyperparameter comparison for Financial Neural Networks models. Canada, 1999–2007.*



▲ **Figure 5.52.** *Canada, 1991–1998.* ***Left:*** *Average pairwise Sharpe ratio difference between all models, single split.* ***Right:*** *Same, but where the evaluation horizon is split into four disjoint subperiods as described in the text.*



▲ **Figure 5.53.** *Canada, 1999–2007.* ***Left:*** *Average pairwise Sharpe ratio difference between all models, single split.* ***Right:*** *Same, but where the evaluation horizon is split into four disjoint subperiods as described in the text.*

5.E Appendix: Detailed European Model Results

Results for the European model follow the same pattern as the United States presented in §5.C/p. 212. In particular, we follow the same methodology of splitting the test into two disjoint periods (1991–1998 and 1999–2007), using the first for hyperparameter selection and the second for (indicative) performance evaluation.

The variable names used in the analyses of variance are described on p. 212.

5.E.1 Linear Regression

Tables 5.50 and 5.51 show the ANOVA tables for the Linear Regression model. All hyperparameters, except for weight decay, have a significant impact on performance. Looking at the pairwise mean differences (Figs. 5.54 and 5.55), the use of monthly returns instead of daily ones proves to be, by a large and significant margin, the best choice; this persists on the validation period. The best input type (fundamental versus all) switches between the validation and test periods. The same lack of consistency affects the rectification threshold.

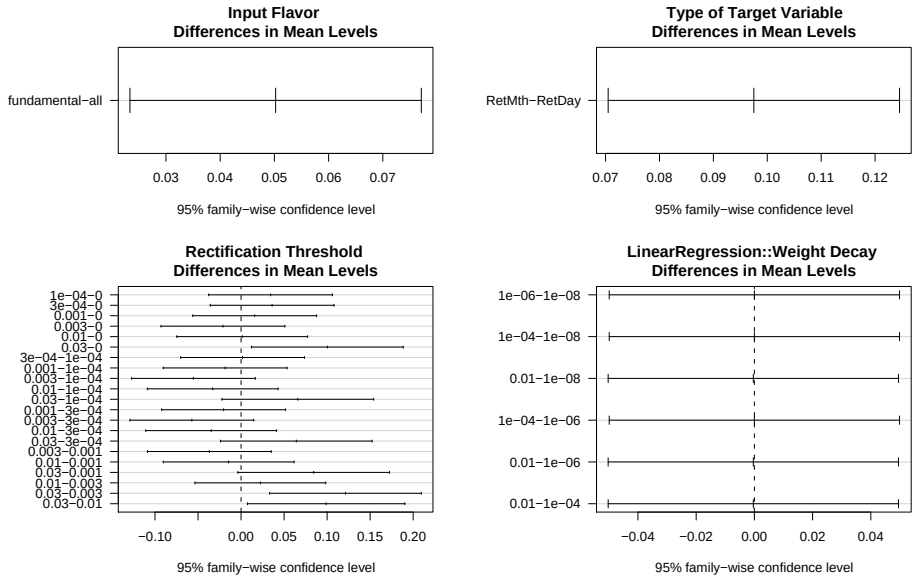
The selected hyperparameters on the 1991–1998 period are:

- `EquityIndexes.input.flavor` = `fundamental`
- `EquityIndexes.rectify.thresh` = 0.03
- `EquityIndexes.target.init` = `RetMth`

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	16.0405	5.3468	283.4725	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.2549	0.2549	13.5129	0.0002702	***
EquityIndexes.rectify.thresh	6	0.3878	0.0646	3.4267	0.0026175	**
EquityIndexes.target.init	1	1.0280	1.0280	54.5002	9.521e-13	***
OptLR.weight.decay	3	1.018e-05	3.394e-06	0.0002	0.9999967	
Residuals	389	7.3373	0.0189			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.50. ANOVA *results for Europe, Linear Regression model, Period ending in 1998.*



▶ **Figure 5.54.** *Hyperparameter comparison for Linear Regression models. Europe, 1991–1998.*

Table 5.51. ANOVA *results for Europe, Linear Regression model, Period ending in 2007.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	2.33526	0.77842	103.2413	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.59936	0.59936	79.4931	< 2.2e-16	***
EquityIndexes.rectify.thresh	6	0.33596	0.05599	7.4264	1.547e-07	***
EquityIndexes.target.init	1	0.85115	0.85115	112.8875	< 2.2e-16	***
OptLR.weight.decay	3	8.334e-08	2.778e-08	3.684e-06	1	
Residuals	385	2.90283	0.00754			

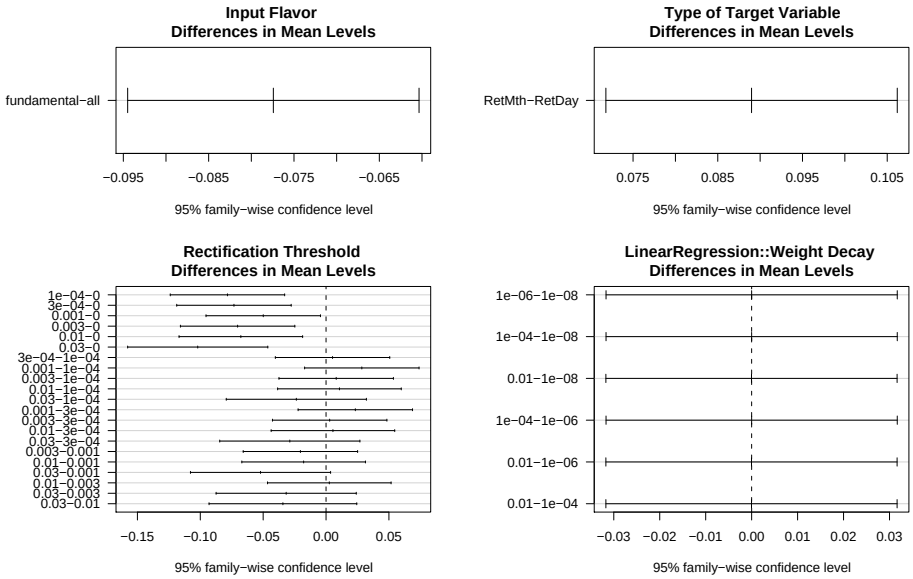
Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

5.E.2 Gaussian Process

Tables 5.52 and 5.53 show the ANOVA tables for the Gaussian Process model. Only the type of target variable shows a significant and consistent impact, whereas the input type is significant only during the test period. Looking at the pairwise mean differences (Figs. 5.56 and 5.57), the `SIGN-DAY` targets clearly dominate during validation period and, although they remain fine, don’t stand out as convincingly.

The selected hyperparameters on the 1991–1998 period are:

- `EquityIndexes.target.init` = `SignDay`



◀ **Figure 5.55.** *Hyperparameter comparison for Linear Regression models. Europe, 1999–2007.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	8.8876	2.9625	85.9878	<2e-16	***
EquityIndexes.input.flavor	1	0.0055	0.0055	0.1609	0.6888	
EquityIndexes.target.init	4	5.3589	1.3397	38.8860	<2e-16	***
optimizer.options.nstages	4	0.0358	0.0089	0.2596	0.9035	
Residuals	187	6.4427	0.0345			

Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

Table 5.52. ANOVA *results for Europe, Gaussian Process model, Period ending in 1998.*

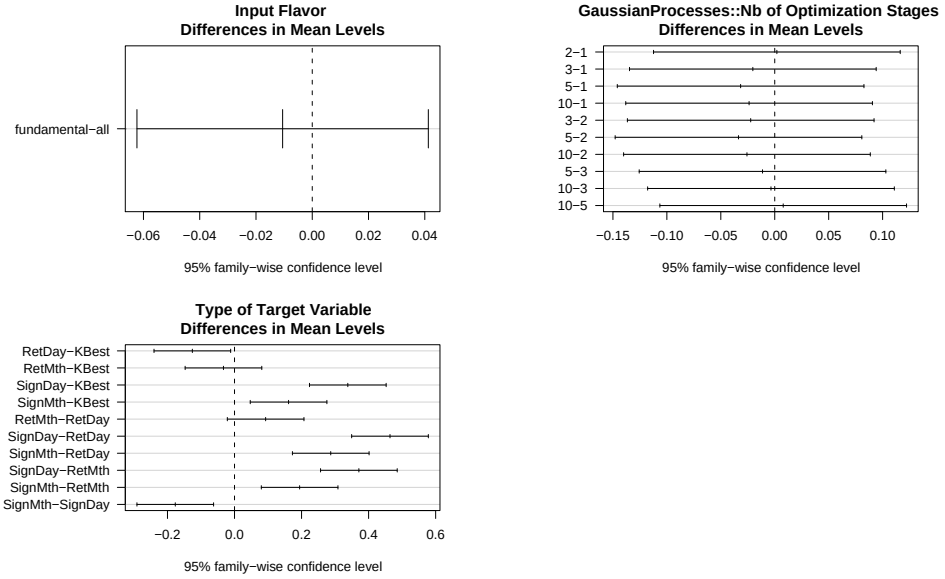
	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	3.0127	1.0042	21.9867	3.061e-12	***
EquityIndexes.input.flavor	1	0.3045	0.3045	6.6670	0.01059	*
EquityIndexes.target.init	4	0.5727	0.1432	3.1344	0.01593	*
optimizer.options.nstages	4	0.0332	0.0083	0.1816	0.94771	
Residuals	187	8.5413	0.0457			

Signif. codes: 0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1

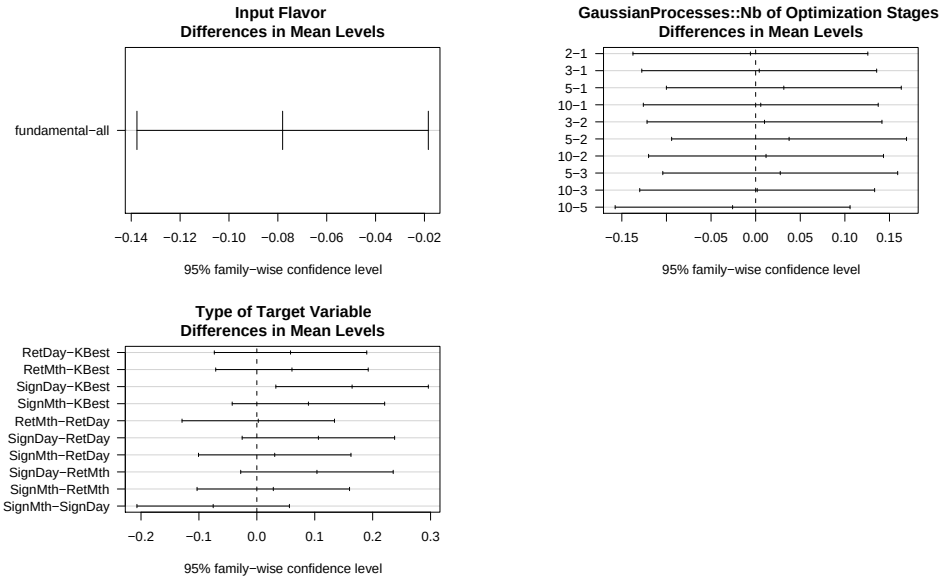
Table 5.53. ANOVA *results for Europe, Gaussian Process model, Period ending in 2007.*

5.E.3 Classification Neural Network

Tables 5.54 and 5.55 show the ANOVA tables for the Classification Neural Network model. The number of hidden units is seen to be consistently significant between the validation and test periods; the other hyperparameters less so. Looking at the pairwise mean differences (Figs. 5.58 and 5.59), there is marked instability in the best hyperparameter values: nearly all



► **Figure 5.56.** *Hyper-parameter comparison for Gaussian Process models. Europe, 1991–1998.*



► **Figure 5.57.** *Hyper-parameter comparison for Gaussian Process models. Europe, 1999–2007.*

the selected values do not remain consistent between the validation and test periods, with the exception of the number of hidden units; particularly distressing is the variability in the impact of weight decay, whose effects (among the tried range) completely reverse.

The selected hyperparameters on the 1991–1998 period are:

- `Classification.nb.classes` $= 3$
- `EquityIndexes.target.init` $\neq$ `KBest`
- `OptClassifNN.weight.decay` $= 10^{-2}$
- `OptClassifNN.nhidden` $= 0$

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	28.8011	9.6004	289.6989	< 2.2e-16	***
Classification.nb.classes	1	0.1209	0.1209	3.6468	0.05668	.
EquityIndexes.input.flavor	1	0.0006	0.0006	0.0184	0.89214	
EquityIndexes.target.init	2	0.1556	0.0778	2.3481	0.09648	.
OptClassifNN.nhidden	3	0.7763	0.2588	7.8085	4.094e-05	***
OptClassifNN.weight.decay	2	0.6471	0.3236	9.7635	6.786e-05	***
Residuals	563	18.6574	0.0331			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.54. ANOVA *results for Europe, Classification Neural Network model, Period ending in 1998.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	4.3112	1.4371	48.3780	< 2.2e-16	***
Classification.nb.classes	1	0.0223	0.0223	0.7512	0.38648	
EquityIndexes.input.flavor	1	0.0911	0.0911	3.0673	0.08043	.
EquityIndexes.target.init	2	3.3737	1.6868	56.7867	< 2.2e-16	***
OptClassifNN.nhidden	3	1.5269	0.5090	17.1338	1.164e-10	***
OptClassifNN.weight.decay	2	0.1598	0.0799	2.6897	0.06877	.
Residuals	563	16.7238	0.0297			

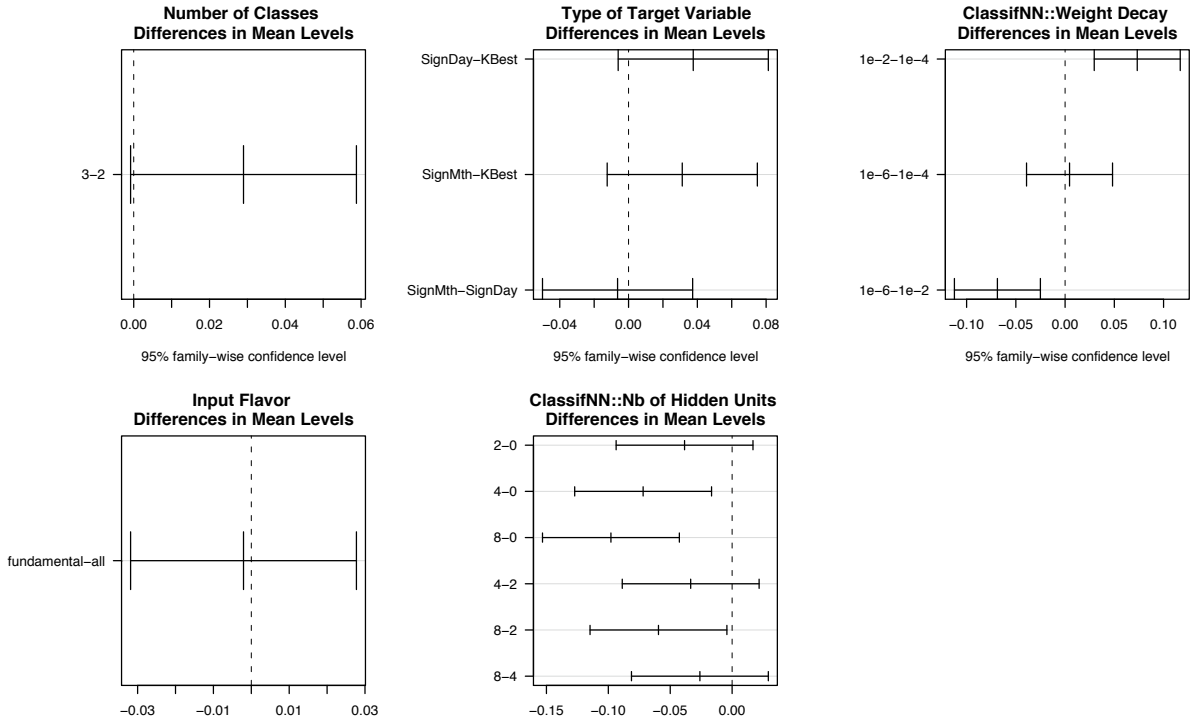
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.55. ANOVA *results for Europe, Classification Neural Network model, Period ending in 2007.*

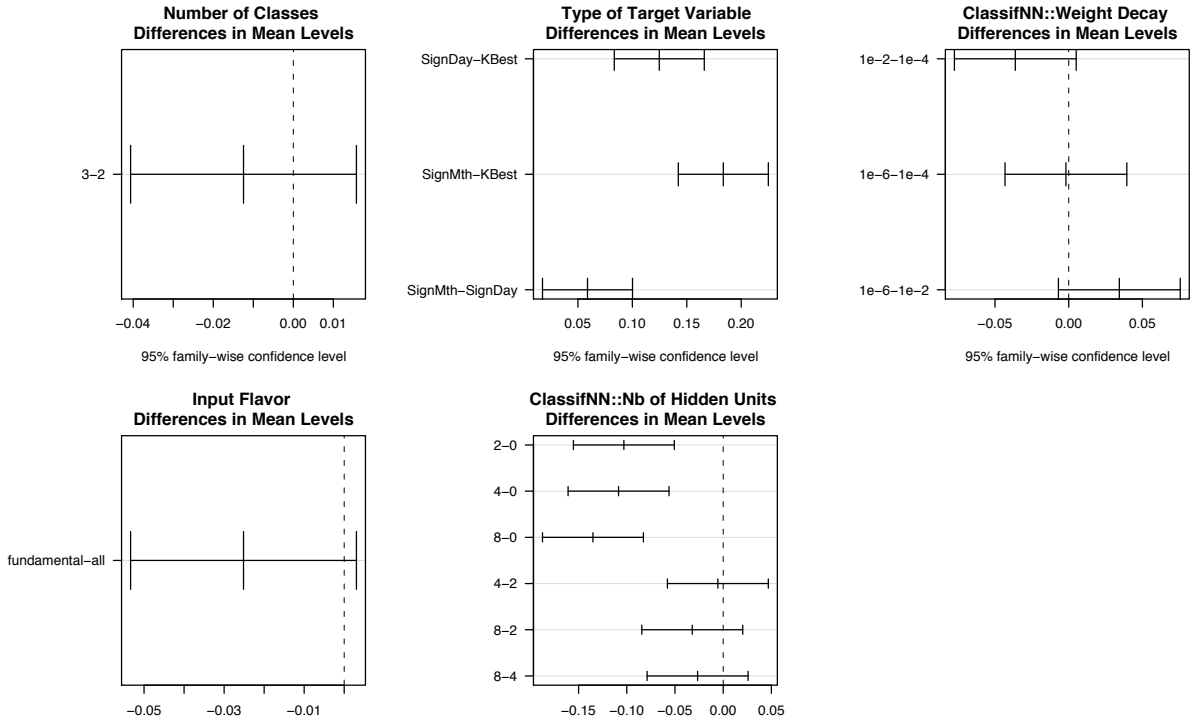
5.E.4 Financial Neural Network

As for the U.S. results, we separate out the analysis between two separate settings: (i) using K-Best targets, and (ii) using monthly and daily targets.Tables 5.56 and 5.57 show the ANOVA tables for the FinancialNNet model **using K-best targets**, whereas Tables 5.58 and 5.59 show them for `SIGN-DAILY` and `SIGN-MONTHLY` targets.

Considering K-best targets alone, the selected hyperparameters on the 1991–1998 period are (the pairwise mean difference plots are omitted for brevity):



▲ **Figure 5.58.** *Hyperparameter comparison for Classification Neural Networks models. Europe, 1991–1998.*



▲ **Figure 5.59.** *Hyperparameter comparison for Classification Neural Networks models. Europe, 1999–2007.*

- `EquityIndexes.target.init` = `KBest`
- `OptFinancialNNet.prefit.nstages` $\neq 0.0$
- `OptFinancialNNet.prop.cost` $= 0.002$
- `OptFinancialNNet.weight.decay` $\neq 10^{-2}$

Considering the *monthly and daily returns* targets, the selected hyperparameters on the 1991–1998 period are:

- `EquityIndexes.target.init` $\neq$ `KBest`
- `OptFinancialNNet.prefit.nstages` $\neq 0.0$
- `OptFinancialNNet.prop.cost` $= 0.002$
- `OptFinancialNNet.weight.decay` $\neq 10^{-2}$

A direct comparison between these two subsets (looking only at differences in the effect of `EquityIndexes.target.init`) indicates that there are no significant difference between the type of target, on neither period.

The pairwise mean differences for the resulting union appear in Figures 5.60 and 5.61. As with previous results (*cf.* Canada), we note the importance of the prefitting stage, as performance degrades significantly and consistently if no prefitting is performed. Moreover, excessive weight decay is seen to consistently impact results. The level of transaction costs in the financial criterion is unambiguous: a small cost (20 BP) works best.

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	67.194	22.398	507.9410	<2e-16	***
EquityIndexes.input.flavor	1	0.085	0.085	1.9284	0.1651	
OpTrProblem.prop.cost	3	0.020	0.007	0.1486	0.9306	
OptFinancialNNet.hint.penalty	1	0.001	0.001	0.0261	0.8716	
OptFinancialNNet.prefit.nstages	2	33.934	16.967	384.7766	<2e-16	***
OptFinancialNNet.prop.cost	2	9.498	4.749	107.7024	<2e-16	***
OptFinancialNNet.weight.decay	2	9.048	4.524	102.5991	<2e-16	***
Residuals	1713	75.536	0.044			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.56. ANOVA *results for Europe, Financial Neural Network model (with K-best targets), Period ending in 1998.*

Table 5.57. ANOVA *results for Europe, Financial Neural Network model (with K-best targets), Period ending in 2007.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	9.231	3.077	72.8736	<2e-16	***
EquityIndexes.input.flavor	1	0.010	0.010	0.2478	0.6187	
OpTrProblem.prop.cost	3	0.040	0.013	0.3145	0.8149	
OptFinancialNNet.hint.penalty	1	0.089	0.089	2.1082	0.1467	
OptFinancialNNet.prefit.nstages	2	16.424	8.212	194.4909	<2e-16	***
OptFinancialNNet.prop.cost	2	7.277	3.639	86.1723	<2e-16	***
OptFinancialNNet.weight.decay	2	15.369	7.684	181.9898	<2e-16	***
Residuals	1713	72.330	0.042			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Table 5.58. ANOVA *results for Europe, Financial Neural Network model (without K-best targets), Period ending in 1998.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	37.326	12.442	268.8709	< 2.2e-16	***
EquityIndexes.input.flavor	1	0.091	0.091	1.9726	0.1605	
EquityIndexes.target.init	1	0.071	0.071	1.5357	0.2156	
OptFinancialNNet.hint.penalty	1	0.015	0.015	0.3248	0.5689	
OptFinancialNNet.prefit.nstages	2	16.651	8.325	179.9092	< 2.2e-16	***
OptFinancialNNet.prop.cost	2	4.875	2.437	52.6697	< 2.2e-16	***
OptFinancialNNet.weight.decay	2	3.335	1.667	36.0318	9.527e-16	***
Residuals	851	39.380	0.046			

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

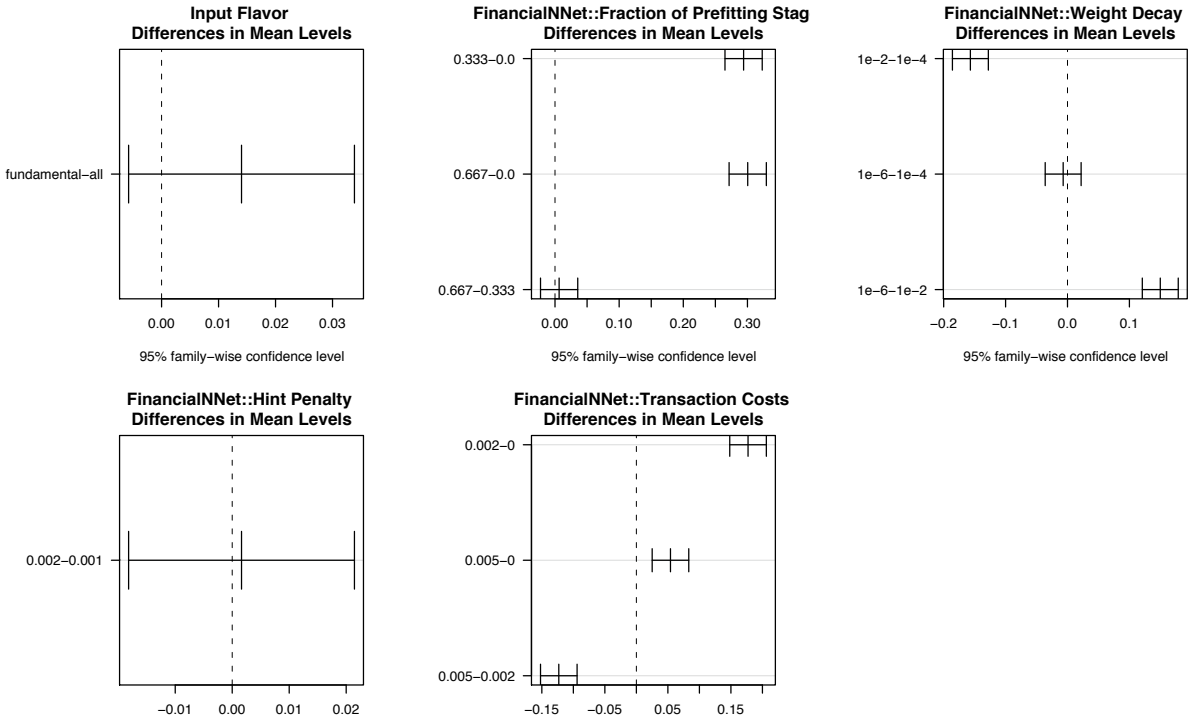
Table 5.59. ANOVA *results for Europe, Financial Neural Network model (without K-best targets), Period ending in 2007.*

	Df	Sum Sq	Mean Sq	F value	Pr(>F)	
Date	3	4.872	1.624	41.0351	<2e-16	***
EquityIndexes.input.flavor	1	0.017	0.017	0.4375	0.5085	
EquityIndexes.target.init	1	0.026	0.026	0.6548	0.4186	
OptFinancialNNet.hint.penalty	1	0.021	0.021	0.5317	0.4661	
OptFinancialNNet.prefit.nstages	2	9.419	4.710	119.0098	<2e-16	***
OptFinancialNNet.prop.cost	2	4.068	2.034	51.3914	<2e-16	***
OptFinancialNNet.weight.decay	2	4.748	2.374	59.9868	<2e-16	***
Residuals	851	33.677	0.040			

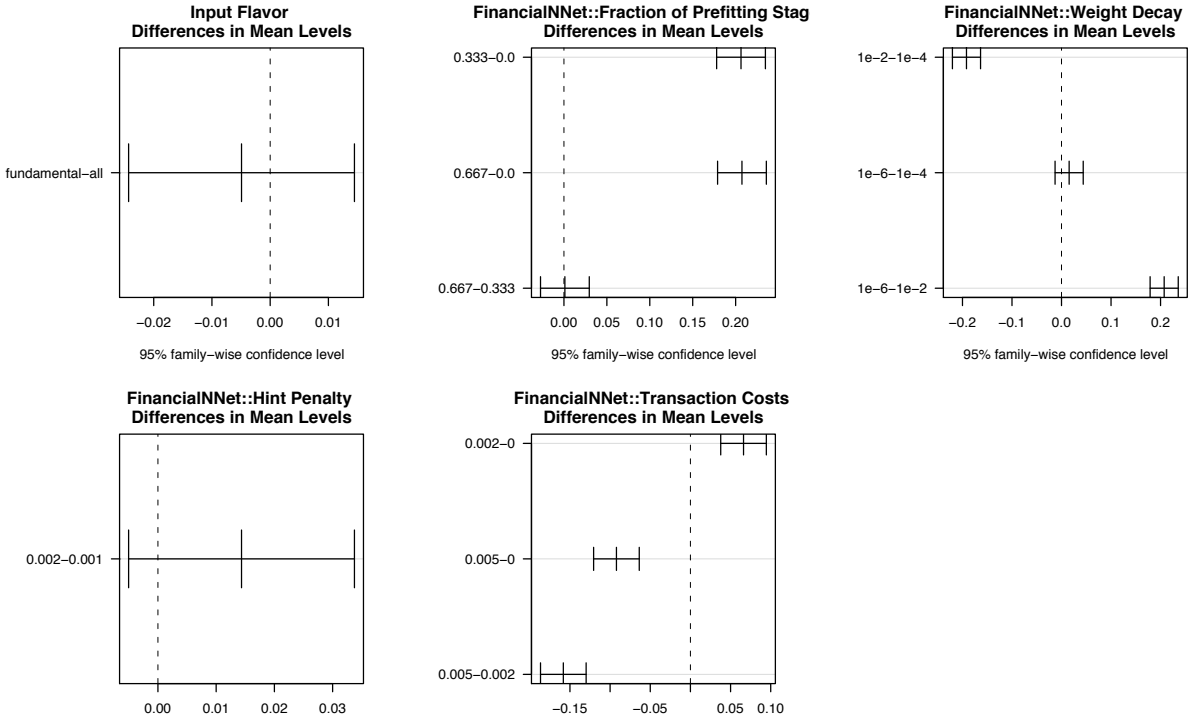
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

5.E.5 Overall Results

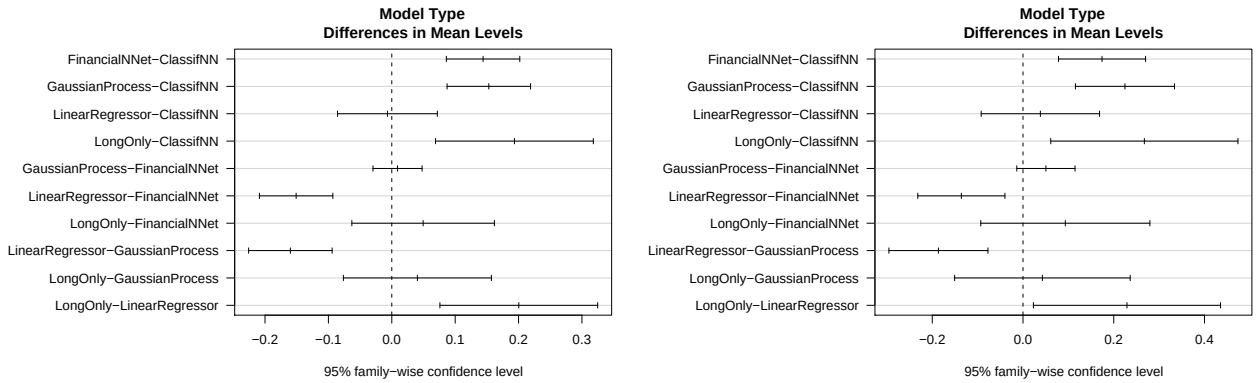
We follow the same procedure as for the U.S. to compare between models, using for each the selected set of hyperparameters. Figures 5.52 and 5.53 illustrate the pairwise mean differences between all models, respectively over the 1991–1998 and 1999–2007 periods. Both the "one-split" and "four-split" results suggest the same relative rankings, during both periods, with the FinancialNNet, Gaussian Process and Long-Only models being roughly equiv-



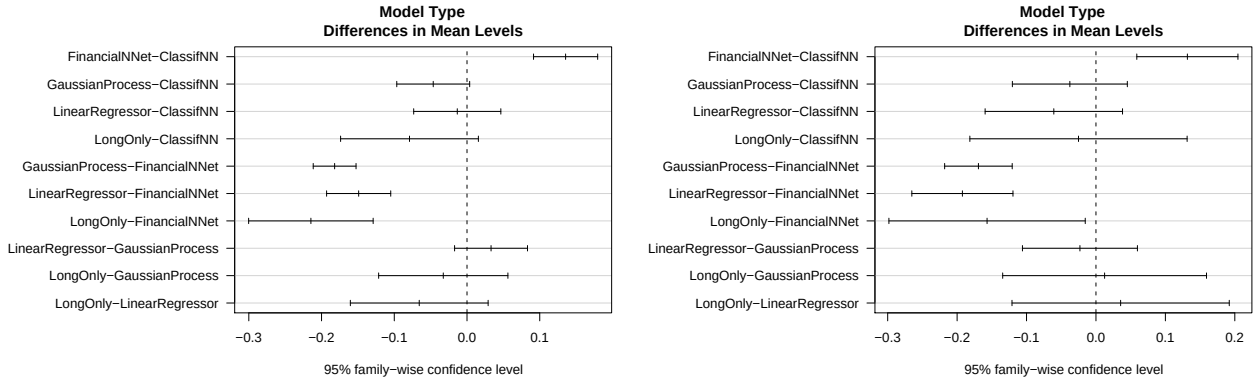
▲ **Figure 5.60.** *Hyperparameter comparison for Financial Neural Networks models. Europe, 1991–1998.*



▲ **Figure 5.61.** *Hyperparameter comparison for Financial Neural Networks models. Europe, 1999–2007.*



▲ **Figure 5.62.** *Europe, 1991–1998.* ***Left:*** *Average pairwise Sharpe ratio difference between all models, single split.* ***Right:*** *Same, but where the evaluation horizon is split into four disjoint subperiods as described in the text.*



▲ **Figure 5.63.** *Europe, 1999–2007.* ***Left:*** *Average pairwise Sharpe ratio difference between all models, single split.* ***Right:*** *Same, but where the evaluation horizon is split into four disjoint subperiods as described in the text.*

alent during the validation period, and the FinancialNNet clearly outperforming during the test period.

6 Augmented Functional Time Series Representation and Forecasting with Gaussian Processes

An economist is an expert who will know tomorrow why the things he predicted yesterday didn't happen today.

— *Laurence J. Peter*

CLASSICAL TIME-SERIES FORECASTING MODELS, such as ARIMA*models (Hamilton 1994), assume that forecasting is performed at a fixed horizon, which is implicit in the model. An overlaying deterministic time trend may be fit to the data, but is generally of fixed and relatively simple functional form (e.g. linear, quadratic, or sinusoidal for periodic data). To forecast beyond the fixed horizon, it is necessary to iterate forecasts in a multi-step fashion. These models are good at representing the short-term dynamics of the time series, but degrade rapidly when longer-term forecasts must be made, usually quickly converging to the unconditional expectation of the process after removal of the deterministic time trend. This is a major issue in applications that require a forecast over a **complete future trajectory**, and not a single (or restricted) horizon. These models are also constrained to deal with regularly-sampled data, and make it difficult to condition the time trend on explanatory variables, especially when iteration of short-term forecasts has to be performed. To a large extent, the same problems are present with non-linear generalizations of such models, such as time-delay or recurrent neural networks (Bishop 1995), which simply allow the short-term dynamics to become nonlinear but leave open the question of forecasting complete future trajectories. Recurrent neural networks can, in principle, model arbitrarily long dependencies in time series, but efficient learning in those models is difficult (Bengio, Simard, and Frasconi 1994).

*Autoregressive Integrated Moving Average.

Functional Data Analysis (FDA) (Ramsay and Silverman 2005) has been proposed in the statistical literature as an answer to some of these concerns. The central idea is to consider a whole curve as an example (specified by a finite number of pairs (t, y_t)), which can be represented by coefficients in a non-parametric basis expansion such as splines. This implies learning about complete trajectories as a function of time, hence the "functional" designation. Since time is viewed as an independent variable, the approach can forecast at arbitrary horizons and handle irregularly-sampled data. Typically, FDA is used without explanatory time-dependent variables, which are

*Earlier versions of this chapter appeared as (Chapados and Bengio 2007a) and (Chapados and Bengio 2008). It has been submitted for publication as (Chapados and Bengio 2009a).

important for the kind of applications we shall be considering. Furthermore, the question remains of how to integrate a progressively-revealed information set in order to make increasingly more precise forecasts of the same future trajectory. To incorporate conditioning information, we consider here the output of a prediction to be a whole forecasting curve (as a function of t).

This chapter presents a solution to the problem of forecasting a complete future trajectory based on the use of *Gaussian Processes* (O'Hagan 1978; Williams and Rasmussen 1996; Rasmussen and Williams 2006). They constitute a general and flexible class of models for nonlinear regression and classification. Over the past decade, they have received wide attention in the machine learning community, having originally been introduced in geostatistics, where they are known under the name "Kriging" (Matheron 1973; Cressie 1993). They differ from usual approaches to feed-forward neural networks in that they engage in a full Bayesian treatment, supplying a complete posterior distribution of forecasts (which are jointly Gaussian). For regression, they are also computationally relatively simple to implement, the basic model requiring only solving a system of linear equations, albeit one of size equal to the number of training examples (requiring $O(N^3)$ computation).

Moreover, a deep connection exists between Gaussian Processes and neural networks: it can be shown that the prior distribution over functions implied by a (Bayesian) one-layer feed-forward neural network tends to a Gaussian Process when the number of hidden units in the network tends to infinity, if a standard isotropic prior over network weights is assumed (Neal 1996).

Motivation

The motivation for this work comes from forecasting and actively trading price spreads between commodity futures contracts (see, e.g., Hull 2005, for an introduction). Since futures contracts expire and have a finite duration, this problem is characterized by the presence of a large number of separate historical time series, which all can be of relevance in forecasting a new time series. For example, we expect seasonalities to affect similarly all the series. Furthermore, conditioning information, in the form of macroeconomic variables, can be of importance, but exhibit the cumbersome property of being released periodically, with explanatory power that varies across the forecasting horizon. In other words, when making a very long-horizon forecast, the model should not incorporate conditioning information in the same way as when making a short- or medium-term forecast. A possible solution to this problem is to have multiple models for forecasting each time series, one for each time scale. However, this is hard to work with, requires a high degree of skill on the part of the modeler, and is not amenable to robust automation when one wants to process hundreds of time series. In addition, in order to measure risk associated with a particular trade (buying at time t and selling at time t'), we need to estimate the *covariance of the price*

predictions associated with these two points in the trajectory.

These considerations motivate the use of Gaussian processes, which naturally provide a covariance matrix between forecasts made at several points. To tackle the challenging task of forecasting and trading spreads between commodity futures, we introduce here a form of functional data analysis in which the function to be forecast is indexed both by the date of availability of the information set and by the forecast horizon. The predicted trajectory is thus represented as a functional object associated with a distribution, a Gaussian process, from which the risk of different trading decisions can readily be estimated. This approach allows incorporating input variables that cannot be assumed to remain constant over the forecast horizon, like statistics of the short-term dynamics.

As an added benefit, the proposed method turns out to be very intuitive to practitioners. The notion of seeing a forecast trajectory for the spread fits much better with the way that traders work in managing trades, and seems more natural than next-period asset expected return and variance of traditional portfolio theory.*

Previous Work

Gaussian processes for time-series forecasting have been considered before. Multi-step forecasts are explicitly tackled by Girard, Rasmussen, Candela, and Murray-Smith (2003), wherein uncertainty about the intermediate values is formally incorporated into the predictive distribution to obtain more realistic uncertainty bounds at longer horizons. However, this approach, while well-suited to purely autoregressive processes, does not appear amenable to the explicit handling of exogenous input variables. Furthermore, it suffers from the restriction of only dealing with regularly-sampled data. Our approach draws from the CO_2 model of Rasmussen and Williams (2006) as an example of application-specific covariance function engineering.

Organization of this Chapter

We start with an overview of known phenomena regarding contract spreads in futures markets (§6.1/p. 246) and review the principal theoretical elements pertaining to Gaussian processes for nonlinear regression (§6.2/p. 247). We then explain in detail the methodology developed for forecasting the complete future trajectory of a spread using Gaussian processes (§6.3/p. 254). We follow with an explanation of the experimental setting for evaluating forecasting performance, including a statistical test that account for cross-correlations introduced by sequential validation (§6.4/p. 261), and an account of the forecasting results of the proposed methodology against several

*Questions that are of daily concern to a trader go beyond the immediate sign of the position — whether *long* or *short* — but cover the validity of the entry and exit points, the expected profit from the trade, the expected timeframe and the conditions for which one should consider an early exit.

benchmark models (§6.5/p. 268). We continue with the elaboration of a criterion to take price trajectory forecasts and turn them in trading decisions (§6.6/p. 268) and financial performance results on a portfolio of 30 spreads (§6.7/p. 271). Finally, §6.8/p. 280 presents directions for future research.

6.1 On Commodity Spreads

Until the work of Working (1949), it had been the norm to consider futures contracts at different maturities on the same underlying commodity as being "substantially independent", the factors impacting one contract (such as expectations of a large harvest in a given month) having little bearing on the others. His *theory of price of storage*, backed by a large body of empirical studies on the behavior of wheat futures (such as, e.g. Working 1934), invalidated the earlier views and established the basis of *intertemporal pricing relationships* that form a key element in understanding the spread behavior of storable commodities.

It has long been recognized that commodity prices, including spreads, can exhibit complex behavior. As Working aptly wrote more than 70 years ago (Working 1935),

> If the important factors bearing on the price of any one commodity were always very few in number and related to the price in very simple fashion, direct multiple correlation analysis might appear entirely adequate, even in light of present general knowledge of its limitations. But intensive realistic study has revealed that, for some commodity prices, the number of factors that must be regarded as really important is rather large. Regressions are frequently curvilinear. The effects of price factors are often not independent, but joint. The factors, or at least the most suitable measures of them, are not known in advance, but remain to be determined; the character of the functional relationships between the factors, separately or jointly, and the price, is unknown and may not safely be assumed linear.

Kim and Leuthold (2000) provide evidence of large shifts in the distributional behavior of corn, live cattle, gold and T-bonds across time periods and temporal aggregation horizon.

Several authors have examined the specific influences of spread seasonalities and other eventual forecastable behavior. Simon (1999) finds evidence of a long-run equilibrium (cointegration) relationship in the soybeans crush spread,* once seasonality and linear trend are accounted for. He also shows mean-reversion at a five-day horizon. Simple (in-sample) trading rules that account for both phenomena show profit after transaction costs. Girma and

* *Which consists in taking a long position in the soybeans contract, and offsetting shorts in both soybean meal and soybean oil.*

Paulson (1998) find significant seasonality at both the monthly and trading-week levels in the petroleum complex spreads* and apply trading rules based on the observed seasonality to extract abnormal out-of-sample returns for the $3:2:1$ crack spread, less so for the simpler spreads. Dutt, Fenton, Smith, and Wang (1997) examined the effects of intra- versus inter-crop year spreads on the volatility of agricultural futures spreads. In the precious metal spreads, Liu and Chou (2003) considered long-run parity relationships between the gold and silver markets and obtain that significant riskless profits could be earned based on the forecasts of an error-correcting model. This market was previously investigated by Wahab, Cohn, and Lashgari (1994), and pure gold spreads were the subject of a study by Poitras (1987). In equity index futures, Butterworth and Holmes (2002) finds some intermarket mispricings between FTSE 100 and FTSE Mid 250 futures, although the possibilities of profitable trading after transaction costs appeared slim. More recently, Dunis, Laws, and Evans (2006c) applied nonlinear methods, such as recurrent neural networks and filter rules, to profitably trade petroleum spreads.

*Collectively referred to as "crack spreads".

In this work we shall consider the simplest type of calendar spreads: intracommodity calendar spreads, obtained by simultaneously taking a long and equal (in terms of the number of contracts) short position in two different maturities of the same underlying commodity. The price of the spread is simply given by the subtraction of the two prices, since they express the same deliverable quantity. Moreover, many of the above papers introducing trading simulations consider the *continuous-contract* spread formed by rolling from one futures contract to another as they reach maturity. As shall be made clear below, a unique property of the methodology proposed in this work is the ability to directly consider the separate histories of the spreads of interest in previous years and use them to forecast the evolution of a spread for the current year. As such, we can dispense with having to specify a rollover policy, since we can directly work with several individual time series without having to somehow merge them into a continuous-contract spread.

6.2 Review of Gaussian Processes for Regression

The present section briefly reviews Gaussian processes at a level sufficient for understanding the spread forecasting methodology developed in the next section.

6.2.1 Basic Concepts for the Regression Case

A Gaussian process is a generalization of the Gaussian distribution: it represents a probability distribution over *functions* which is entirely specified by a mean and covariance *functions*. Borrowing the succinct definition from

Rasmussen and Williams (2006), we formally define a Gaussian process (GP) as

Definition 11 (Gaussian process) *A* Gaussian process *is a collection of random variables, any finite number of which have a joint Gaussian distribution.*

Let $\mathbf{x}$ index into the real process $f(\mathbf{x})$.* We write

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\cdot), k(\cdot,\cdot)\right), \tag{6.1}$$

where functions $m(\cdot)$ and $k(\cdot,\cdot)$ are, respectively, the mean and covariance functions:

$$\begin{aligned} m(\mathbf{x}) &= \mathbb{E}\left[f(\mathbf{x})\right], \\ k(\mathbf{x}_1, \mathbf{x}_2) &= \mathbb{E}\left[\left(f(\mathbf{x}_1) - m(\mathbf{x}_1)\right)\left(f(\mathbf{x}_2) - m(\mathbf{x}_2)\right)\right]. \end{aligned}$$

In a regression setting, we shall assume that we are given a training set $\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid i = 1, \ldots, N\}$, with $\mathbf{x}_i \in \mathbb{R}^D$ and $y_i \in \mathbb{R}$. For convenience, let $\mathbf{X}$ be the matrix of all training inputs, and $\mathbf{y}$ the vector of targets. We shall further assume that the y_i are noisy measurements from the underlying process $f(\mathbf{x})$:

$$y_i = f(\mathbf{x}_i) + \varepsilon_i, \qquad \text{with } \varepsilon_i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_n^2).$$

Regression with a GP is achieved by means of Bayesian inference in order to obtain a posterior distribution over functions given a suitable prior and training data. Then, given new *test inputs*, we can use the posterior to arrive at a *predictive distribution* conditional on the test inputs and the training data. The predictive distribution is normal for a GP. Although the idea of manipulating distributions over functions may appear cumbersome, the *consistency property* of GPs means that any finite number of values sampled from the process f are jointly normal; hence inference over random functions can be shown to be completely equivalent to inference over a finite number of random variables.

It is often convenient, for simplicity, to assume that the GP prior distribution has a mean of zero,

$$f(\mathbf{x}) \sim \mathcal{GP}(0, k(\cdot,\cdot)).$$

Let $\mathbf{f}' = [f(\mathbf{x}_1)', \ldots, f(\mathbf{x})'_N]$ be the vector of (latent) function values at the training inputs. Their prior distribution is given by

$$\mathbf{f} \sim \mathcal{N}(\mathbf{0}, K(\mathbf{X}, \mathbf{X})),$$

*Contrarily to many treatments of stochastic processes, there is no necessity for $\mathbf{x}$ to represent time; indeed, in our application of Gaussian processes, some of the elements of $\mathbf{x}$ have a temporal interpretation, but most are general macroeconomic variables used to condition the targets.

where $K(\mathbf{X}, \mathbf{X})_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j)$ is matrix formed by evaluating the covariance function between all pairs of training points. (This matrix is also known as the kernel or Gram matrix.) We shall consider the joint prior distribution between the training and an additional set of *test points*, with locations given by the matrix $\mathbf{X}_*$, and whose function values $\mathbf{f}_*$ we wish to infer. Under the GP prior, we have*

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X}, \mathbf{X}) & K(\mathbf{X}, \mathbf{X}_*) \\ K(\mathbf{X}_*, \mathbf{X}) & K(\mathbf{X}_*, \mathbf{X}_*) \end{bmatrix}\right). \tag{6.2}$$

We obtain the predictive distribution at the test points as follows. From Bayes' theorem, the joint posterior given the training data is

$$\begin{aligned} P(\mathbf{f}, \mathbf{f}_* \,|\, \mathbf{y}) &= \frac{P(\mathbf{y} \,|\, \mathbf{f}, \mathbf{f}_*) P(\mathbf{f}, \mathbf{f}_*)}{P(\mathbf{y})} \\ &= \frac{P(\mathbf{y} \,|\, \mathbf{f}) P(\mathbf{f}, \mathbf{f}_*)}{P(\mathbf{y})}, \end{aligned} \tag{6.3}$$

where $P(\mathbf{y} \,|\, \mathbf{f}, \mathbf{f}_*) = P(\mathbf{y} \,|\, \mathbf{f})$ since, by assumption, the likelihood is conditionally independent of $\mathbf{f}_*$ given $\mathbf{f}$, and

$$\mathbf{y} \,|\, \mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma_n^2 I_N) \tag{6.4}$$

with I_N the $N \times N$ identity matrix. The desired predictive distribution is obtained by marginalizing out the training set latent variables,

$$\begin{aligned} P(\mathbf{f}_* \,|\, \mathbf{y}) &= \int P(\mathbf{f}, \mathbf{f}_* \,|\, \mathbf{y}) d\mathbf{f} \\ &= \frac{1}{P(\mathbf{y})} \int P(\mathbf{y} \,|\, \mathbf{f}) P(\mathbf{f}, \mathbf{f}_*) d\mathbf{f}. \end{aligned}$$

Since these distributions are all normal, the result of the (normalized) marginal is also normal, and can be shown to have mean and covariance given by (see §A.4/p. 299 for a derivation)

$$\mathbb{E}\left[\mathbf{f}_* \,|\, \mathbf{y}\right] = K(\mathbf{X}_*, \mathbf{X}) \boldsymbol{\Lambda}^{-1} \mathbf{y}, \tag{6.5}$$

$$\mathrm{Cov}\left[\mathbf{f}_* \,|\, \mathbf{y}\right] = K(\mathbf{X}_*, \mathbf{X}_*) - K(\mathbf{X}_*, \mathbf{X}) \boldsymbol{\Lambda}^{-1} K(\mathbf{X}, \mathbf{X}_*), \tag{6.6}$$

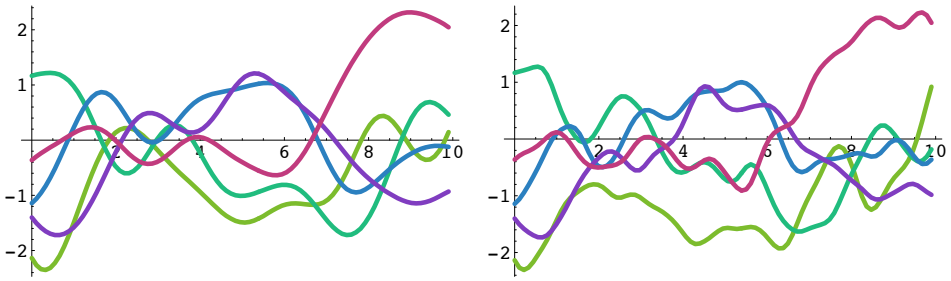
where we have set

$$\boldsymbol{\Lambda} = K(\mathbf{X}, \mathbf{X}) + \sigma_n^2 I_N. \tag{6.7}$$

Computation of $\boldsymbol{\Lambda}^{-1}$ is the most computationally expensive step in Gaussian process regression, requiring $O(N^3)$ time and $O(N^2)$ space.

Note that a natural outcome of GP regression is an expression for not only the expected value at the test points (6.5), but also the full covariance matrix between those points (6.6). We shall be making use of this covariance matrix later.

*Note that the following expressions in this section are implicitly conditioned on the training and test inputs, respectively $\mathbf{X}$ and $\mathbf{X}_*$. The explicit conditioning notation is omitted for brevity.



▶ **Figure 6.1.** ***Left:*** *Random functions drawn from the GP prior with the* ***squared exponential*** *covariance ($\sigma_\ell = 1$).* ***Right:*** *"Same functions" under the* ***rational quadratic*** *covariance ($\sigma_\ell = 1, \alpha = \frac{1}{2}$).*

6.2.2 Choice of Covariance Function

We have so far been silent on the form that the covariance function $k(\cdot, \cdot)$ should take. A proper choice for this function is important for encoding prior knowledge about the problem; several striking examples are given in Rasmussen and Williams (2006). In order to yield valid covariance matrices, the covariance function should, at the very least, be symmetric and positive semi-definite, which implies that all its eigenvalues are nonnegative,

$$\int k(\mathbf{u}, \mathbf{v}) f(\mathbf{u}) f(\mathbf{v})\, d\mu(\mathbf{u})\, d\mu(\mathbf{v}) \geq 0,$$

for all functions f defined on the appropriate space and measure μ.

Two common choices of covariance functions are the *squared exponential**,

*Also known as the Gaussian or radial basis function kernel.

$$k_{\mathrm{SE}}(\mathbf{u}, \mathbf{v}; \sigma_\ell) = \exp\left(-\frac{\|\mathbf{u} - \mathbf{v}\|^2}{2\sigma_\ell^2}\right) \tag{6.8}$$

and the *rational quadratic*

$$k_{\mathrm{RQ}}(\mathbf{u}, \mathbf{v}; \sigma_\ell, \alpha) = \left(1 + \frac{\|\mathbf{u} - \mathbf{v}\|^2}{2\alpha\sigma_\ell^2}\right)^{-\alpha}. \tag{6.9}$$

In both instances, the hyperparameter σ_ℓ governs the *characteristic length-scale* of the covariance function, indicating the degree of smoothness of the underlying random functions. The rational quadratic can be interpreted as an infinite mixture of squared exponentials with different length-scales; it converges to a squared exponential with characteristic length-scale σ_ℓ as $\alpha \to \infty$.

Figure 6.1 illustrates several functions drawn from the GP prior, respectively with the squared exponential and rational quadratic covariance functions. The same random numbers have served for generating both sets, so the functions are "alike" in some sense. Observe that a function under the squared exponential prior is smoother than the corresponding function under the rational quadratic prior.

6.2.3 Optimization of Hyperparameters

Most covariance functions, including those presented above, have free parameters — termed *hyperparameters* — that control their shape, for instance the characteristic length-scale σ_n^2. It remains to explain how their value should be set. This is an instance of the general problem of *model selection*. For many machine learning algorithms, this problem has often been approached by minimizing a validation error through cross-validation (Stone 1974) and such methods have been proposed for Gaussian processes (Sundararajan and Keerthi 2001).

An alternative approach, quite efficient for Gaussian processes, consists in maximizing the *marginal likelihood** of the observed data with respect to the hyperparameters. This function can be computed by introducing latent function values that are immediately marginalized over. Let θ the set of hyperparameters that are to be optimized; and let $K_{\mathbf{X}}(\theta)$ the covariance matrix computed by a covariance function whose hyperparameters are θ,

*Also called the integrated likelihood or (Bayesian) evidence.

$$K_{\mathbf{X}}(\theta)_{i,j} = k(\mathbf{x}_i, \mathbf{x}_j; \theta).$$

The marginal likelihood (here making explicit, for clarity, the dependence on the training inputs and hyperparameters) can be written

$$p(\mathbf{y} \,|\, \mathbf{X}, \theta) = \int p(\mathbf{y} \,|\, \mathbf{f}, \mathbf{X}) p(\mathbf{f} \,|\, \mathbf{X}, \theta)\, d\mathbf{f}, \tag{6.10}$$

where the distribution of observations $p(\mathbf{y} \,|\, \mathbf{f}, \mathbf{X})$, given by eq. (6.4), is conditionally independent of the hyperparameters given the latent function $\mathbf{f}$. Under the Gaussian process prior (see eq. (6.2)), we have $\mathbf{f} \,|\, \mathbf{X}, \theta \sim \mathcal{N}(0, K_{\mathbf{X}}(\theta))$, or in terms of log-likelihood,

$$\log p(\mathbf{f} \,|\, \mathbf{X}, \theta) = -\frac{1}{2}\mathbf{f}' K_{\mathbf{X}}^{-1}(\theta)\mathbf{f} - \frac{1}{2}\log |K_{\mathbf{X}}(\theta)| - \frac{N}{2}\log 2\pi. \tag{6.11}$$

Since both distributions in eq. (6.10) are normal, the marginalization can be carried out analytically to yield

$$\log p(\mathbf{y} \,|\, \mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}'(K_{\mathbf{X}}(\theta)+\sigma_n^2 I_N)^{-1}\mathbf{y} - \frac{1}{2}\log \left|K_{\mathbf{X}}(\theta) + \sigma_n^2 I_N\right| - \frac{N}{2}\log 2\pi. \tag{6.12}$$

This expression can be maximized numerically, for instance by a conjugate gradient algorithm (e.g. Bertsekas 2000) to yield the selected hyperparameters:

$$\theta^* = \arg\max_{\theta} \log p(\mathbf{y} \,|\, \mathbf{X}, \theta).$$

The likelihood function is in general non-convex and this maximization only finds a local maximum in parameter space; however, empirically it usually works very well for a large class of covariance functions, including those covered in §6.2.2/p. 250.

It should be mentioned that completely a Bayesian treatment would not be satisfied with such an optimization, since it only picks a single value for each hyperparameter. Instead, one would define a prior distribution on hyperparameters (an *hyperprior*), $p(\theta)$, and marginalize with respect to this distribution. However, for many "reasonable" hyperprior distributions, this is computationally expensive, and often yields no marked improvement over simple optimization of the marginal likelihood (MacKay 1999).

The gradient of the marginal log-likelihood with respect to the hyperparameters — necessary for numerical optimization algorithms — can be expressed as

$$\frac{\partial \log p(\mathbf{y} \mid \mathbf{X}, \theta)}{\partial \theta_i} = \frac{1}{2}\mathbf{y}' K_{\mathbf{X}}^{-1}(\theta) \frac{\partial K_{\mathbf{X}}(\theta)}{\partial \theta_i} K_{\mathbf{X}}^{-1}(\theta)\mathbf{y} - \frac{1}{2}\mathrm{Tr}\left(K_{\mathbf{X}}^{-1}(\theta) \frac{\partial K_{\mathbf{X}}(\theta)}{\partial \theta_i} \right). \tag{6.13}$$

See Rasmussen and Williams (2006) for details on the derivation of this equation.

6.2.4 Two-Step Training

Since hyperparameter optimization involves a large number of repetitions of solving a linear system, it requires a significantly greater computational effort than the single solution of eq. (6.5) and (6.6). To make this optimization tractable, yet not relinquish our ability to consider enough historical data to obtain a model with adequate predictive power, the experiments of this work (§6.4/p. 261) rely on a *two-step training* procedure:

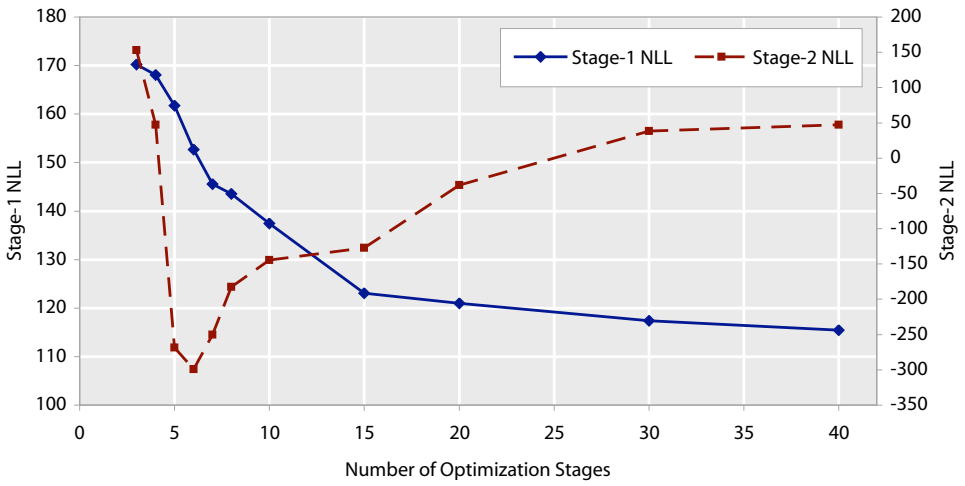
1. First, hyperparameters are optimized on a moderate-sized training set ($N = 500$).

2. Then, *keeping hyperparameters fixed*, we train the final model with a larger training-set ($N = 2000$–3000); our experiments, described below, used $N = 2250$.

Some care must be exercised with this procedure, since it may be subject to the problem of *overfitting* that traditionally plagues neural networks (§3.1.3/p. 73): in the present context, the hyperparameters can overfit if too large a number of optimization stages is used in the first step. This phenomenon is illustrated in Fig. 6.2: both the first-step and second-step NLL* are plotted as a function of the number of optimization stages used in first-step training (hyperparameter optimization). Whereas, as expected, first-step NLL decreases in a monotonic way as a function of the number of stages, the second-step NLL first decreases, reaches a minimum, and starts to increase again, sign that hyperparameters have been too heavily tuned to the peculiarities of the smaller first-step training set.

*Negative Log-Likelihood.

Although Gaussian processes (and Bayesian methods, in general) are sometimes advertised for their robustness to overfitting, this example illustrates that one must constantly remain on guard.



◀ **Figure 6.2.** *Overfitting phenomenon observed during two-step Gaussian Process training. As the number of first-step optimization stages is increased, first-step NLL decreases monotonically, but second-step NLL starts to increase beyond an optimal stage.*

6.2.5 Parameterization for Positive Hyperparameters

It is common for covariance function hyperparameters to be constrained to be positive. For instance, all hyperparameters in the $k_{\text{AUG-RQ}}$ function to be introduced in eq. (6.17) require this constraint. To guarantee positivity, it is occasionally advised* to optimize parameters in the log-domain, i.e. that the final hyperparameters should be obtained as the exponential of parameters that we are actually optimizing.

However, practical experience with this parameterization reveals that it is extremely numerically unstable when used with more than about ten hyperparameters on moderately-sized training sets ($N \gtrapprox 1000$): a single conjugate gradient step may cause the final parameters to become so large (after exponentiation) as to yield, for instance, covariance matrices that are no longer strictly positive-definite.

Instead of optimizing in the log-domain, we have had much success optimizing in the *inverse-softplus domain*, where the final hyperparameters result from taking the softplus function (see eq. (5.5), p. 128) of parameters actually subject to optimization. Since the softplus function has bounded derivatives, it does not exhibit the problems otherwise encountered with the exponential parameterization.

6.2.6 Weighting the Importance of Input Variables: Automatic Relevance Determination

The covariance functions introduced in eq. (6.8) and (6.9) rely on an isotropic Euclidean norm as the similarity measure between two vectors in input space. These functions assume that a global characteristic length-scale σ_ℓ^2 governs proximity evaluation in all input dimensions. Even after normal-

*For instance, see C.E. Rasmussens's publicly available Matlab code for Gaussian processes at `www.gaussianprocess.org`.

izing the input variables to the same scale*, differing predictive strength and noise level among the input variables may make it compelling to use a *specific characteristic length-scale* for each input.

As can be done, for instance, by standardization, i.e. subtracting the mean of each variable and dividing by its standard deviation; see §6.3.3/p. 259.

This amounts to a simple rewriting of the Euclidean norm in the previously-introduced covariance functions as weighted norms, with a weight $\frac{1}{2}\sigma_i^2, i = 1, \ldots, D$ associated to each input dimension. For instance, the squared exponential kernel (eq. 6.8) would take the form

$$k_{\text{SE ARD}}(\mathbf{u}, \mathbf{v}; \sigma) = \exp\left(-\sum_{i=1}^{D} \frac{(\mathbf{u}_i - \mathbf{v}_i)^2}{2\sigma_i^2}\right). \tag{6.14}$$

The hyperparameters σ_i are found by numerically maximizing the marginal likelihood of the prior, as described in the previous section. After optimization, inputs that are found to have little importance are given a high σ_i, so that their importance in the norm is diminished. This procedure is an application to Gaussian processes of *automatic relevance determination*, a soft input variable selection procedure originally proposed in the context of Bayesian neural networks (MacKay 1994; Neal 1996). We revisit this topic in §6.3.5/p. 260 when describing the specific form of the covariance function used in our forecasting methodology.

6.3 Forecasting Methodology

We consider a set of N real time series each of length M_i, $\{y_t^i\}, i = 1, \ldots, N$ and $t = 1, \ldots, M_i$. In our application each i represents a different year, and the series is the sequence of commodity spread prices during the period where it is traded. The lengths of all series are not necessarily identical, but we shall assume that the time periods spanned by the series are "comparable" (e.g. the same range of days within a year if the series follow an annual cycle) so that knowledge from past series can be transferred to a new one to be forecast.

Note that there is no attempt to represent the whole history as a single continuous time series. Rather, each "trading year" of data† is treated as a separate time series,‡ and the year number is used as a continuous independent variable in the regression model. This representation is natural for spreads whose existence, like those of the underlying futures contracts, is tied to specific delivery months, and whose behavior is intimately driven (for agricultural commodities) by seasonalities such as the prevailing crop conditions in a given year. A synthetic illustration of the data representation is shown in Figure 6.3. This should be contrasted to some spread modeling methodologies (e.g. Dunis, Laws, and Evans 2006b) that attempt to create a

†*Which may cross calendar years depending on the maturity months of the spread being modeled.*

‡Nothing in this treatment precludes individual spread price trajectories to span a longer than one year horizon, but for simplicity we shall use the "year" as the unit of trajectory length in the proceeds.

continuous spread series by defining a rollover policy over futures contracts.

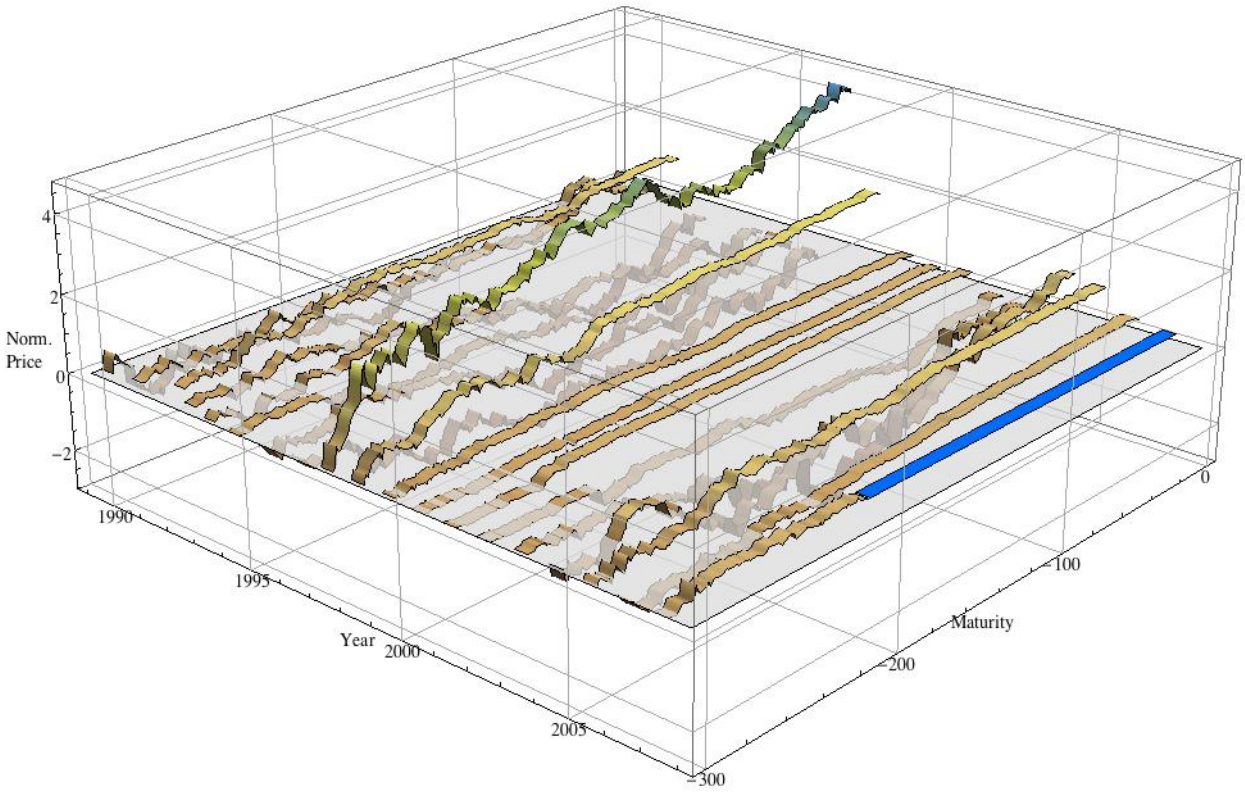
The **forecasting problem** is that given observations from the complete series $i = 1, \ldots, N-1$ and from a *partial last series*, $\{y_t^N\}, t = 1, \ldots, M_N$, we want to extrapolate the last series until a predetermined endpoint, i.e. characterize the joint distribution of $\{y_\tau^N\}, \tau = M_N+1, \ldots, M_N+H$. We are also given a set of non-stochastic explanatory variables specific to each series, $\{\mathbf{x}_t^i\}$, where $\mathbf{x}_t^i \in \mathbb{R}^d$. Our objective is to find an effective representation of $P(\{y_\tau^N\}_{\tau=M_N+1,\ldots,M_N+H} \mid \{\mathbf{x}_t^i, y_t^i\}_{t=1,\ldots,M_i}^{i=1,\ldots,N})$, with τ, i and t ranging, respectively over the forecasting horizon, the available series and the observations within a series.

6.3.1 Functional Representation for Forecasting

In the spirit of functional data analysis, a straightforward attempt at solving the spread evolution forecasting problem is to formulate it as one of regression from a "current date" (along with additional exogenous variables) to spread price. Contrarily to most traditional stationary time-series model that would represent a finite-horizon series return, we include a *representation of the current date as an independent variable*, and regress on (appropriately normalized) spread prices. More specifically, we split the date input into two parts: the current time-series identity i (an integer representing, e.g., the spread trading year) and the time t within the series (we use the number of calendar days remaining until spread maturity, where maturity is defined as that of the near-term spread leg). This yields the model

$$\begin{aligned} \mathbb{E}\left[y_t^i \middle| \mathcal{I}_{t_0}^i\right] &= f(i, t, \mathbf{x}_{t|t_0}^i) \\ \mathrm{Cov}\left[y_t^i, y_{t'}^{i'} \middle| \mathcal{I}_{t_0}^i\right] &= g(i, t, \mathbf{x}_{t|t_0}^i, i', t', \mathbf{x}_{t'|t_0}^{i'}), \end{aligned} \tag{6.15}$$

these expressions being conditioned on the *information set* $\mathcal{I}_{t_0}^i$ containing information up to time t_0 of series i (we assume that all prior series $i' < i$ are also included in their entirety in $\mathcal{I}_{t_0}^i$). From a practical standpoing, we allow all the information in $\mathcal{I}_{t_0}^i$ to be part of the model training set. The notation $\mathbf{x}_{t|t_0}^i$ denotes a forecast of $\mathbf{x}_t^i$ given information available at t_0. Functions f and g result from Gaussian process training, eq. (6.5) and (6.6), using information in $\mathcal{I}_{t_0}^i$. To extrapolate over the unknown horizon, one simply evaluates f and g with the series identity index i set to N and the time index t within a series ranging over the elements of τ (forecasting period). Owing to the smoothness properties of an adequate covariance function, one can expect the last time series (whose starting portion is present in the training data) to be smoothly extended, with the Gaussian process borrowing from prior series, $i < N$, to guide the extrapolation as the time index reaches far enough beyond the available data in the last series. Figure 6.4 illustrates the approach. Note that we have additional input variables (on top of current year, maturity, and maturity delta described below), which are detailed in section 6.4.



▲ **Figure 6.3.** *Illustration of the regression variables around which the forecasting problem is specified. Shown is the price history of the March–July old-crop/new-crop Wheat spread, as a function of the crop year and number of days to maturity. Prices are normalized to start at zero at maturity* -300 *and scaled to unit standard deviation over the sample. To aid the interpretation of price movement, prices that fall below the cutting plane at zero are shown slightly shaded ("under water"). The objective of the forecasting model is to fill out the "blue strip" in the last year of data, given the partial trajectory observed so far for that last year, and the complete trajectories in previous years.*

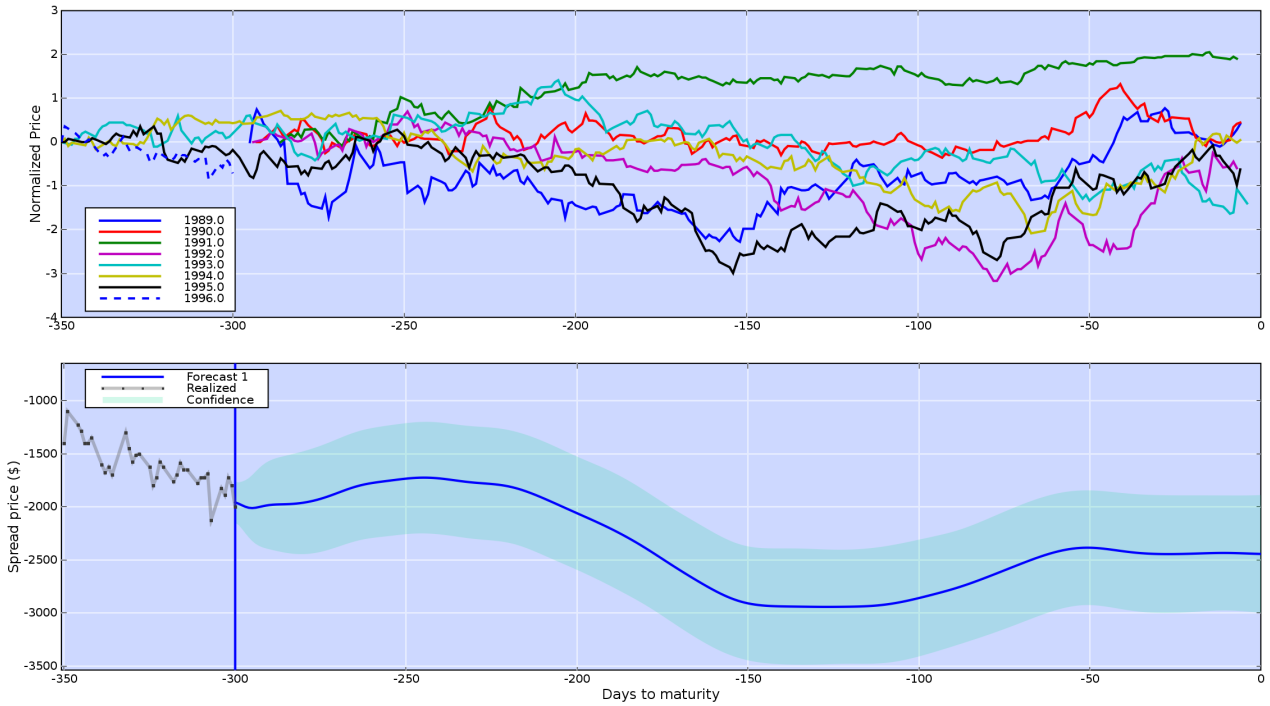
One could worry that including a time representation as independent variables (for instance, the year index, the number of days to maturity, as well as the additional time inputs described in the next section) would lead the Gaussian process to overfit to particular years. In practice, this overfitting does not happen, mainly because time is not a single continuous variable (which would make extrapolation quite difficult), but rather the time representation is explicitly split out across dimensions that are natural for generalization. In the case of seasonal commodities, we expect, *a priori*, similar events to repeat across crop years. Hence, specifying the number of days to maturity as input allows inherent generalization along the dimension of crop growth during a season. Likewise, the inclusion of a separate year index lets the model represent trends that develop over several years, for instance a spread becoming progressively steeper with each crop season.

However, the principal difficulty with this method resides in handling the exogenous inputs $\mathbf{x}^N_{t|t_0}$ over the forecasting period: the realizations of these variables, $\mathbf{x}^N_t$, are not usually known at the time the forecast is made and must be extrapolated with some reasonableness. For slow-moving variables that represent a "level" (as opposed to a "difference" or a "return"), one can conceivably keep their value constant to the last known realization across the forecasting period. However, this solution is restrictive, problem-dependent, and precludes the incorporation of short-term dynamical variables (e.g. the first differences over the last few time-steps) if desired.

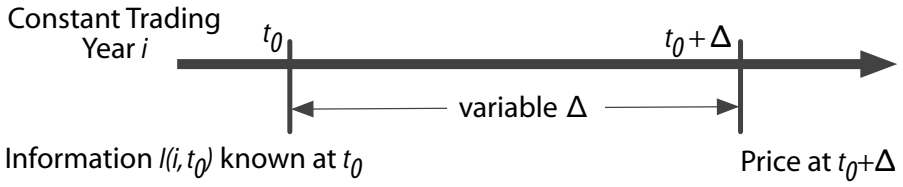
6.3.2 Augmented Functional Representation

We propose in this work to augment the functional representation with an additional input variable that expresses the time *at which* the forecast is being made, in addition to the time *for which* the forecast is made. We shall denote the former the *operation time* and the latter the *target time*. The distinction is as follows: **operation time** represents the time at which the other input variables are observed and the time at which, conceptually, a forecast of the entire future trajectory is performed. In contrast, **target time** represents time at a point of the predicted target series (beyond operation time), given the information known at the operation time. Figure 6.5 illustrates the concept.

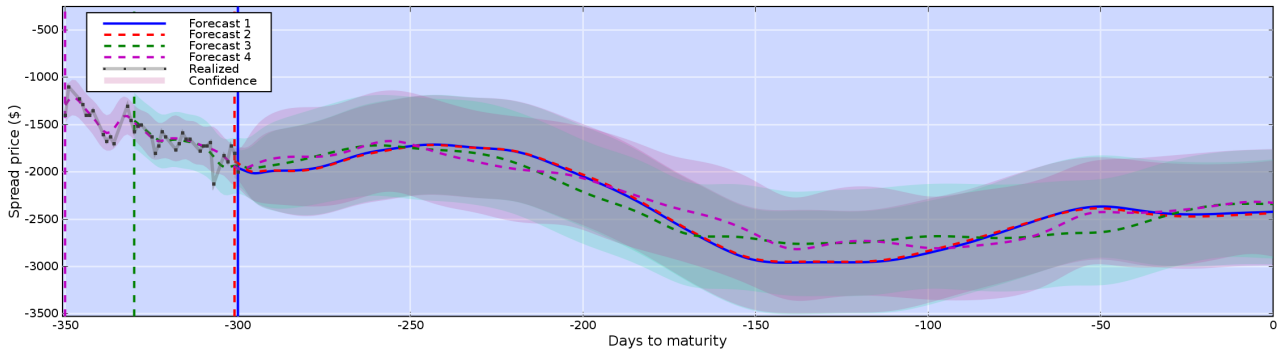
The training set used to form the augmented representation conceptually includes all pairs $\langle \textit{operation time}, \textit{target time} \rangle$ for a given trajectory. As previously, the time series index i remains part of the inputs. In this framework, forecasting is performed by holding the time series index constant to N, the operation time constant to the time M_N of the last observation, the other input variables constant to their last-observed values $\mathbf{x}^N_{M_N}$, and *varying the target time* over the forecasting period τ. Since we are not attempting to extrapolate the exogenous inputs beyond their intended range of validity, this approach admits general input variables, without restriction as to their type, and whether they themselves can be forecast.



▲ **Figure 6.4.** ***Top part:*** *Training set for March–July Wheat spread, containing data until 1995/05/19; each spread year constitutes a distinct trajectory. The x-axis is the negative number of days to maturity (so that time flows to the right). The y-axis is normalized spread price, where the normalization procedure is described in the text. The partial trajectory observed so far for the current year (1996) is the dashed line.* ***Bottom part:*** *Forecasts made by extending the partial spread trajectory of the current year using the Gaussian process regression methodology. The y-axis is the actual spread price in \$ (where individual contract prices are multiplied by a contract size of 50 for Wheat).*



► **Figure 6.5.** *The augmented functional representation attempts to capture the relationship between the price at the target time $t_0 + \Delta$ and information available at the operation time t_0, for all t_0 and Δ within a given spread trading year.*



▲ **Figure 6.6.** *Forecasts made from several operation times, given the same training set as in Figure 6.4.*

It can be convenient to represent the target time as a positive difference Δ from the operation time t_0. In contrast to eq. (6.15), this yields the representation

$$\begin{aligned} \mathbb{E}\left[y^i_{t_0+\Delta}\middle|\mathcal{I}^i_{t_0}\right] &= f(i,t_0,\Delta,\mathbf{x}^i_{t_0}) \\ \operatorname{Cov}\left[y^i_{t_0+\Delta}, y^{i'}_{t'_0+\Delta'}\middle|\mathcal{I}^i_{t_0}\right] &= g(i,t_0,\Delta,\mathbf{x}^i_{t_0},i',t'_0,\Delta',\mathbf{x}^{i'}_{t'_0}), \end{aligned} \tag{6.16}$$

where we have assumed the operation time to coincide with the end of the information set. As previously, functions f and g are the outcome of Gaussian process training (eq. (6.5) and (6.6)) using all information in $\mathcal{I}^i_{t_0}$. In particular, the g function represents the posterior covariance between an arbitrary pair of points, possibly belonging to different time series (when $i \neq i'$); this is precisely what allows information to be "transferred" across past historical time series and allows learning of seasonal patterns. The exact form of the covariance function used in the Gaussian process is discussed in §6.3.5/p. 260.

It is significant to note that this augmentation—the explicit separation between the operation and the target time—allows to dispense with the problematic extrapolation $\mathbf{x}^i_{t|t_0}$ of the inputs, instead allowing a direct use of the last available values $\mathbf{x}^i_{t_0}$. Moreover, from a given information set, nothing precludes forecasting the same trajectory from several operation times $t' < t_0$, which can be used as a means of evaluating the stability of the obtained forecast. This idea is illustrated in Figure 6.6.

6.3.3 Input and Target Variable Preprocessing

Input variables are subject to minimal preprocessing before being provided as input to the gaussian process: we standardize them to zero mean and unit standard deviation. The price targets require additional treatment: since the price level of a spread can vary significantly from year to year, we normalize the price trajectories to *start at zero* at the start of every year,

by subtracting the first price. Furthermore, in order to get slightly better behaved optimization, we divide the price targets by their overall standard deviation.

6.3.4 Training Set Subsampling

A major practical concern with Gaussian processes is their ability to scale to larger problems. As seen in §6.2.1/p. 247, the basic training step of Gaussian process regression involves the inversion of an $N \times N$ matrix (where N is the number of training examples), or the equivalent solution of a linear system. As such, training times can be expected to scale as $O(N^3)$. In practice, more than one such step is required: the optimization of hyperparameters described in §6.2.3/p. 251 involves the numerical optimization of a marginal likelihood function, requiring repeated solutions of same-sized problems — from several dozen to a few hundred times, depending on the number of hyperparameters and the accuracy sought.

The problem is compounded by augmentation, which requires, in principle, a greatly expanded training set. In particular, the training set must contain sufficient information to represent the output variable for the *many combinations of operation and target times* that can be provided as input. In the worst case, this implies that the number of training examples grows quadratically with the length of the training time series. In practice, a downsampling scheme is used wherein only a fixed number of target-time points is sampled for every operation-time point.*

A large number of approximation methods have been proposed to tackle large training sets; see Quiñonero-Candela and Rasmussen (2005) and Rasmussen and Williams (2006) for good surveys of the field. Most approaches involve some form of subsampling of the training set, used in conjunction with a revised form of the estimators that can account for the influence of excluded training examples. Our downsampling scheme amounts to the simplest approximation method, called *subset of data*, which consists simply in taking a subset of the original examples.

6.3.5 Covariance Function Engineering

One of the most significant attractions of Gaussian processes lies in their ability to tailor their behavior for a specific application through the choice of covariance function. It is even possible to create completely novel functions, as long as they satisfy the positive semi-definiteness requirements of a valid covariance function (see §6.2.2/p. 250), or construct new ones according to straightforward composition rules (Shawe-Taylor and Cristianini 2004; Rasmussen and Williams 2006) — a process often called *covariance function (*or *kernel) engineering*.

*This number was 15 in our experiments, and these were not regularly spaced, with longer horizons spaced farther apart. Furthermore, the original daily frequency of the data was reduced to keep approximately one operation-time point per week.

In the present application, we use a modified form of the *rational quadratic* covariance function with hyperparameters for automatic relevance determination , which is expressed as

$$k_{\text{AUG-RQ}}(\mathbf{u}, \mathbf{v}; \ell, \alpha, \sigma_f, \sigma_{\text{TS}}) = \sigma_f^2 \left(1 + \frac{1}{2\alpha} \sum_{k=1}^{d} \frac{(\mathbf{u}_k - \mathbf{v}_k)^2}{\ell_k^2}\right)^{-\alpha} + \sigma_{\text{TS}}^2 \delta_{i_{\mathbf{u}}, i_{\mathbf{v}}}, \tag{6.17}$$

where $\delta_{j,k} \triangleq I[j = k]$ is the Kronecker delta. The variables $\mathbf{u}$ and $\mathbf{v}$ are values in the augmented representation introduced previously, containing the three variables representing time (current time-series index or year, operation time, target time) as well as the additional explanatory variables. The notation $i_{\mathbf{u}}$ denotes the index of the time-series component of input variable $\mathbf{u}$.

The last term of the covariance function, the Kronecker delta, is used to induce an increased similarity among points that belong to the same time series (e.g. the same spread trading year). By allowing a series-specific average level to be maintained into the extrapolated portion, the presence of this term was found to bring better forecasting performance.

The hyperparameters $\ell_i, \alpha, \sigma_f, \sigma_{\text{TS}}, \sigma_n$ are found by maximizing the marginal likelihood on the training set by a standard conjugate gradient optimization, as outlined in §6.2.3/p. 251.

As is standard practice with Gaussian processes, separate hyperparameters (σ_f^2 and σ_{TS}^2) are estimated for the main and delta portion of the covariance function. In principle, a single one would be necessary to correctly estimate the process mean. However, having separate hyperparameters allows comparison among several alternative covariance functions on the marginal likelihood criterion.

6.4 Experimental Setting

To establish the benefits of the proposed functional representation for forecasting commodity spread prices, we compared it against other likely models on three common grain and grain-related spreads:* the Soybeans (January–July, July–November, July–September, August–November, August–March), Soybean Meal (May–September July–December, July–September, August–December, August–March), and Soft Red Wheat (March–July, March–September, May–December, May–September, July–December). The forecasting

*Our convention is to first give the *short leg* of the spread, followed by the *long leg*. Hence, Soybeans 1–7 should be interpreted as taking a short position (i.e. selling) in the January Soybeans contract and taking an offsetting long (i.e. buying) in the July contract. Traditionally, intra-commodity spread positions are taken so as to match the number of contracts on both legs — the number of short contracts equals the number of long ones — not the dollar value of the long and short sides.

task is to *predict the complete future trajectory* of each spread (taken individually), from 200 days before maturity until maturity.

6.4.1 Methodology

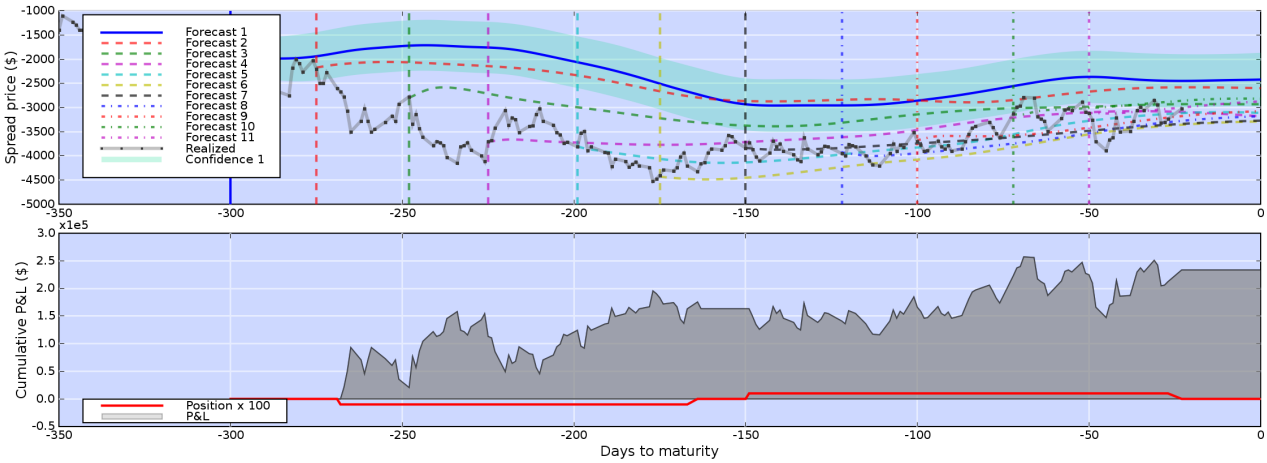
Realized prices in the previous trading years are provided from 250 days to maturity, using data going back to 1989. The first test year is 1994. Within a given trading year, the time variables represent the number of calendar days to maturity of the near leg; since no data is observed on week-ends, training examples are sampled on an irregular time scale.

Performance evaluation proceeds through a *sequential validation* procedure (see Chapter 4): within a trading year, we first train models 200 days before maturity and obtain a first forecast for the future price trajectory. We then retrain models every 25 days, and obtain revised portions of the remainder of the trajectory; note that within a given trading year, a complete forecast until the end of the year is carried not, not a "one-step-ahead" forecast. Proceeding sequentially, this operation is repeated for succeeding trading years. All forecasts are compared amongst models on squared-error and negative log-likelihood criteria (see "assessing significance", below).

6.4.2 Models Compared

The "complete" model to be compared against others is based on the augmented-input representation Gaussian process with the modified rational quadratic covariance function eq. (6.17). In addition to the three variables required for the representation of time, the following inputs were provided to the model: (i) the current spread price and the price of the three nearest futures contracts on the underlying commodity term structure, (ii) economic variables (the stock-to-use ratio and year-over-year difference in total ending stocks) provided on the underlying commodity by the U.S. Department of Agriculture (U.S. Department of Agriculture 2008). This model is denoted **AugRQ/all-inp**. An example of the sequence of forecasts made by this model, repeated every 25 times steps, is shown in the upper panel of Figure 6.7.

One may wonder at what would appear to be systematic biases in the first few forecasts made by the model (bearing in mind that the forecast plotted in a given color only uses information up to the vertical line of the same color on the plot). This behavior is related to generalization across years: given that very little information from the current trading year is available to the early forecasts, more is necessarily borrowed from earlier years, including mean spread levels. However, from about 225 days prior to maturity and beyond (towards zero), the model adjusts its forecasts to account for the now-larger history of current-year behavior, and the bias disappears. This progressive and automatic adjustment of forecasts depending on the information in current versus previous years can be seen as a strong advantage of this methodology.



▲ **Figure 6.7.** ***Top Panel:*** *Illustration of multiple forecasts, repeated every 25 days, of the 1996 March–July Wheat spread (dashed lines); realized price is in gray. Although the first forecast (smooth solid blue, with confidence bands) mistakes the overall price level, it approximately correctly identifies local price maxima and minima, which is sufficient for trading purposes.* ***Bottom Panel:*** *Position taken by the trading model (in red: short, then neutral, then long), and cumulative profit of that trade (gray).*

To determine the value added by each type of input variable, we include in the comparison two models based on exactly on the same architecture, but providing less inputs: **AugRQ/less-inp** does not include the economic variables. **AugRQ/no-inp** further removes the price inputs, leaving only the time-representation inputs. Moreover, to quantify the performance gain of the augmented representation of time, the model **StdRQ/no-inp** implements a "standard time representation" that would likely be used in a functional data analysis model; as described in eq. (6.15), this uses a single time variable instead of splitting the representation of time between the operation and target times.

Finally, we compare against simpler models: **Linear/all-inp** uses a dot-product covariance function to implement Bayesian linear regression, using the full set of input variables described above. And **AR(1)** is a simple linear autoregressive model. For this last model, the predictive mean and covariance matrix are established as follows (see, e.g. Hamilton 1994). We consider the scalar data generating process

$$y_t = \phi\, y_{t-1} + \varepsilon_t, \qquad \varepsilon_t \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma^2), \tag{6.18}$$

where the process $\{y_t\}$ has an unconditional mean of zero.* The h-step ahead forecast from time t under this model, which we write $y_{t+h|t}$, is obtained by iterating h times eq. (6.18),

$$y_{t+h|t} = \phi^h y_t + \sum_{i=0}^{h-1} \phi^i \varepsilon_{t+h-i}. \tag{6.19}$$

**In practice, we subtract the empirical mean on the training set before proceeding with the analysis.*

The minimum mean-squared point forecast, $\hat{y}_{t+h|t}$, is given by the conditional expectation given information available at time t, $\mathcal{I}_t$,

$$\hat{y}_{t+h|t} = \mathbb{E}\left[y_{t+h} \,|\, \mathcal{I}_t\right] = \phi^h y_t. \tag{6.20}$$

The conditional variance of $y_{t+h|t}$ is given by

$$\operatorname{Var}\left[y_{t+h} \,|\, \mathcal{I}_t\right] = \sum_{i=0}^{h-1} \phi^{2i}\sigma^2 = \sigma^2 \frac{\phi^{2h}-1}{\phi^2-1}. \tag{6.21}$$

To evaluate the covariance between two forecasts, respectively at h and h' steps ahead, we start by evaluating the conditional expectation of the product,

$$\begin{aligned}
\mathbb{E}\left[y_{t+h|t}\, y_{t+h'|t} \,|\, \mathcal{I}_t\right] &= \mathbb{E}\left[\left(\phi^h y_t + \sum_{i=0}^{h-1}\phi^i \varepsilon_{t+h-i}\right)\left(\phi^{h'} y_t + \sum_{i=0}^{h'-1}\phi^i \varepsilon_{t+h'-i}\right)\middle|\, \mathcal{I}_t\right] \\
&= \phi^{h+h'} y_t^2 + \mathbb{E}\left[\sum_{i=0}^{h-1}\phi^{h-i-1}\varepsilon_{t+i}\sum_{j=0}^{h'-1}\phi^{h'-j-1}\varepsilon_{t+j}\middle|\, \mathcal{I}_t\right] \\
&= \phi^{h+h'} y_t^2 + \sigma^2 \sum_{i=0}^{M}\phi^{h+h'-2i-2} \\
&= \phi^{h+h'} y_t^2 + \sigma^2 \phi^{h+h'} \frac{1-\phi^{-2M}}{\phi^2-1},
\end{aligned}$$

with $M \triangleq \min(h, h')$. The covariance is given as

$$\begin{aligned}
\operatorname{Cov}\left[y_{t+h|t}, y_{t+h'|t} \,|\, \mathcal{I}_t\right] &= \mathbb{E}\left[y_{t+h|t}\, y_{t+h'|t} \,|\, \mathcal{I}_t\right] - \mathbb{E}\left[y_{t+h|t} \,|\, \mathcal{I}_t\right]\mathbb{E}\left[y_{t+h'|t} \,|\, \mathcal{I}_t\right] \\
&= \phi^{h+h'} y_t^2 + \sigma^2 \phi^{h+h'} \frac{1-\phi^{-2M}}{\phi^2-1} - \left(\phi^h y_t\right)\left(\phi^{h'} y_t\right) \\
&= \sigma^2 \phi^{h+h'} \frac{1-\phi^{-2M}}{\phi^2-1}.
\end{aligned} \tag{6.22}$$

Equations (6.20) and (6.22) are used in conjunction with the approach outlined in §6.6/p. 268 to make trading decisions.

It should be noted that the iterated one-step linear forecaster is optimal at longer horizons only when the model is properly specified, i.e. the model matches the DGP* (Clements and Hendry 1998). Otherwise, a direct forecasting at the desired horizon is preferable. However, due to complexity of estimating a large number of concurrent models in order to forecast the entire spread trajectory at all possible future horizons, and *especially the computation of a reliable covariance matrix among them*, we settled on using the iterated one-step model as our "naïve" benchmark.

*Data Generating Process.

6.4.3 Significance of Forecasting Performance Differences

For each trajectory forecast, we measure the squared error (SE) made at each time-step along with the negative log-likelihood (NLL) of the realized price under the predictive distribution. To account for differences in target variable distribution throughout the years, we normalize the SE by dividing it by the standard deviation of the test targets in a given year. Similarly, we normalize the NLL by subtracting the likelihood of a univariate Gaussian distribution estimated on the test targets of the year.

Due to the serial correlation it exhibits, the time series of performance differences (either SE or NLL) between two models cannot directly be subjected to a standard t-test of the null hypothesis of no difference in forecasting performance. The well-known Diebold-Mariano test (Diebold and Mariano 1995) corrects for this correlation structure in the case where a *single time series* of performance differences is available. This test is usually expressed as follows. Let $\{d_t\}$ be the sequence of *error differences* between two models to be compared. Let $\bar{d} = \frac{1}{M}\sum_t d_t$ be the mean difference. The sample variance of $\bar{d}$ is readily shown (Diebold and Mariano 1995) to be

$$\hat{v}_{\mathrm{DM}} \triangleq \mathrm{Var}[\bar{d}] = \frac{1}{M}\sum_{k=-K}^{K}\hat{\gamma}_k,$$

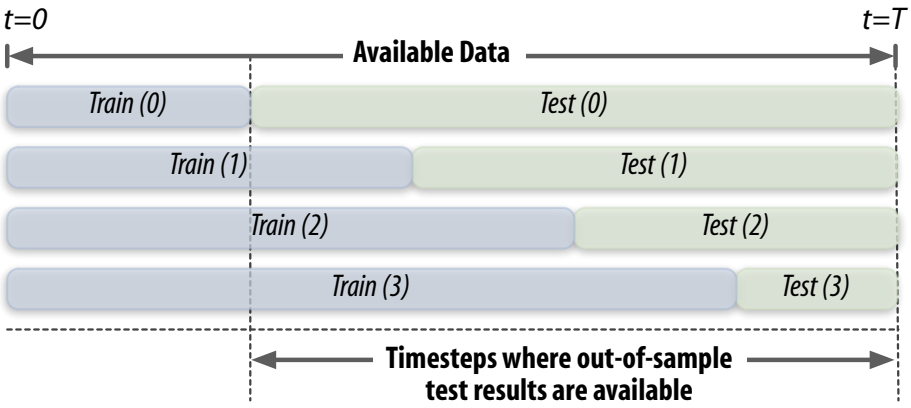
where M is the sequence length and $\hat{\gamma}_k$ is an estimator of the lag-k autocovariance of the d_ts. The maximum lag order K is a parameter of the test and must be determined empirically. Then the statistic $DM = \bar{d}/\sqrt{\hat{v}_{\mathrm{DM}}}$ is asymptotically distributed as $\mathcal{N}(0,1)$ and a classical test of the null hypothesis $\bar{d} = 0$ can be performed.

Unfortunately, even the Diebold-Mariano correction for autocorrelation is not sufficient to compare models in the present case. Due to the repeated forecasts made for the same time-step across *several iterations of sequential validation*, the error sequences are likely to be *cross-correlated* since they result from models estimated on strongly overlapping training sets. This effect is illustrated in Figure 6.8.

Although generally omitted, the steps in the derivation of the Diebold-Mariano variance estimator are useful to understand in order to generalize to the case of sequential validation. In order to perform an hypothesis test, we need an estimator of the variance of the sample mean in the case where the elements exhibit correlation. We have

$$\begin{aligned}\mathrm{Var}[\hat{d}] &= \frac{1}{N^2}\mathrm{Var}\left[\sum_i d_i\right] \\ &= \frac{1}{N^2}\sum_i\sum_j \mathrm{Cov}[d_i, d_j]. \end{aligned} \tag{6.23}$$

Diebold and Mariano work under the hypothesis of stationarity and maxi-



▶ **Figure 6.8.** *Illustration of* ***sequential validation*** *with strongly overlapping test sets. A model is retrained at regular intervals, each time tested on the remainder of the data. At each iteration, parts of the test data from the previous iteration is added to the training set.*

mum lag order of the covariances, such that

$$\mathrm{Cov}[d_i, d_j] = \begin{cases} \gamma_{|i-j|} & \text{if } |i-j| \leq K, \\ 0 & \text{if } |i-j| > K. \end{cases} \tag{6.24}$$

They then consider a *typical row* i of the covariance matrix, and notice that for this row, the inner summation in eq. (6.23) is equal to the sum of the covariance spectrum, $\sum_{k=-K}^{K} \gamma_k$. They then posit that this inner summation is repeated *for every row of the covariance matrix* (even on the boundaries), presumably to offset the fact that the higher-order correlations (greater than K) have been dropped. Substituting in (6.23), we have

$$\begin{aligned} \mathrm{Var}[\hat{d}] &= \frac{1}{N^2} \sum_i \sum_{k=-K}^{K} \gamma_k \\ &= \frac{1}{N} \sum_{k=-K}^{K} \gamma_k, \end{aligned}$$

where the last line follows from the inner summation not depending on the outer. This is the Diebold-Mariano variance estimator.

Generalization for Sequential Validation For simplicity, we shall assume only two iterations of sequential validation, with the following elements resulting from testing. Test sets 1 and 2 overlap at timesteps 4 to 6:

Test set #1	1	2	3	4	5	6
Test set #2				4	5	6

Assuming covariance stationarity and keeping a maximum lag order $K = 2$, we obtain the following form for the covariance matrix linking all elements

$$\left(\begin{array}{cccccc|ccc}
\gamma_0^1 & \gamma_1^1 & \gamma_2^1 & & & & & & \\
\gamma_1^1 & \gamma_0^1 & \gamma_1^1 & \gamma_2^1 & & & & & \\
\gamma_2^1 & \gamma_1^1 & \gamma_0^1 & \gamma_1^1 & \gamma_2^1 & & & & \\
 & \gamma_2^1 & \gamma_1^1 & \gamma_0^1 & \gamma_1^1 & \gamma_2^1 & \gamma_0^{1,2} & \gamma_1^{1,2} & \gamma_2^{1,2} \\
 & & \gamma_2^1 & \gamma_1^1 & \gamma_0^1 & \gamma_1^1 & \gamma_1^{1,2} & \gamma_0^{1,2} & \gamma_1^{1,2} \\
 & & & \gamma_2^1 & \gamma_1^1 & \gamma_0^1 & \gamma_2^{1,2} & \gamma_1^{1,2} & \gamma_0^{1,2} \\
\hline
 & & & \gamma_0^{1,2} & \gamma_1^{1,2} & \gamma_2^{1,2} & \gamma_0^2 & \gamma_1^2 & \gamma_2^2 \\
 & & & \gamma_1^{1,2} & \gamma_0^{1,2} & \gamma_1^{1,2} & \gamma_1^2 & \gamma_0^2 & \gamma_1^2 \\
 & & & \gamma_2^{1,2} & \gamma_1^{1,2} & \gamma_0^{1,2} & \gamma_2^2 & \gamma_1^2 & \gamma_0^2
\end{array}\right)$$

where γ_k^i denote the lag-k autocovariances within test set i, and $\gamma_k^{i,j}$ denote the lag-k cross-covariances between test sets i and j. The horizontal and vertical lines have been set between the elements belonging to test boundaries.

The above matrix explicly shows what is the impact of the cross-covariances (the off-diagonal blocks) on the resulting variance. Our extension to the Diebold-Mariano test consists in simply estimating those terms and incorporating them into the overall variance estimator, yielding the *cross-covariance-corrected Diebold-Mariano variance estimator*

$$\hat{v}_{\mathrm{CCC-DM}} = \frac{1}{M^2}\left(\sum_i M_i \sum_{k=-K}^{K} \hat{\gamma}_k^i + \sum_i \sum_{j \neq i} M_{i \cap j} \sum_{k=-K'}^{K'} \hat{\gamma}_k^{i,j}\right), \qquad (6.25)$$

where M_i is the number of examples in test set i, $M = \sum_i M_i$ is the total number of examples, $M_{i \cap j}$ is the number of time-steps where test sets i and j overlap, $\hat{\gamma}_k^i$ denote the estimated lag-k autocovariances within test set i, and $\hat{\gamma}_k^{i,j}$ denote the estimated lag-k cross-covariances between test sets i and j. The maximum lag order for cross-covariances, K', is possibly different from K (our experiments used $K = K' = 15$).* This revised variance estimator was used in place of the usual Diebold-Mariano statistic in the results presented below.

*Empirically, the results are fairly insensitive to the precise choice K and K', as long as values greater than about 8 are used.

6.5 Forecasting Performance Results

Results of the forecasting *performance difference* between **AugRQ/all-inp** and all other models is shown in Table 6.1. We observe that **AugRQ/all-inp** generally beats the others on both the SE and NLL criteria, often statistically significantly so. In particular, the augmented representation of time is shown to be of value (i.e. comparing against **StdRQ/no-inp**). Moreover, the Gaussian process is capable of making good use of the additional price and economic input variables, although not always with the traditionally accepted levels of significance.

Figure 6.9 illustrates the same results graphically, and makes it easier to see, at a glance, how **AugRQ/all-inp** performs against other models.

A more detailed investigation of absolute and relative forecast errors of each model, on the March–July Wheat spread and for every year of the 1994–2007 period, is presented in appendix (§6.9/p. 281).

6.6 From Forecasts to Trading Decisions

We applied this forecasting methodology based on an augmented representation of time to trading a portfolio of spreads. Within a given trading year, we apply an information-ratio criterion to greedily determine the best trade into which to enter, based on the entire price forecast (until the end of the year) produced by the Gaussian process. More specifically, let $\{p_t\}$ be the future prices forecast by the model at some operation time (presumably the time of last available element in the training set). The expected forecast dollar profit of buying at t_1 and selling at t_2 is simply given by $\tilde{p}_{t_2} - \tilde{p}_{t_1}$, where $\tilde{p}_{t_i}$ is the present-value discounted price: $\tilde{p}_{t_i} = e^{-r_f m_i} p_{t_i}$, with $m_i = t_i - t_0$ the number of days for which to discount and r_f a risk-free rate.

Of course, a prudent investor would take trade risk into consideration. A simple approximation of risk is given by the trade profit volatility. This yields the *forecast information ratio** of the trade

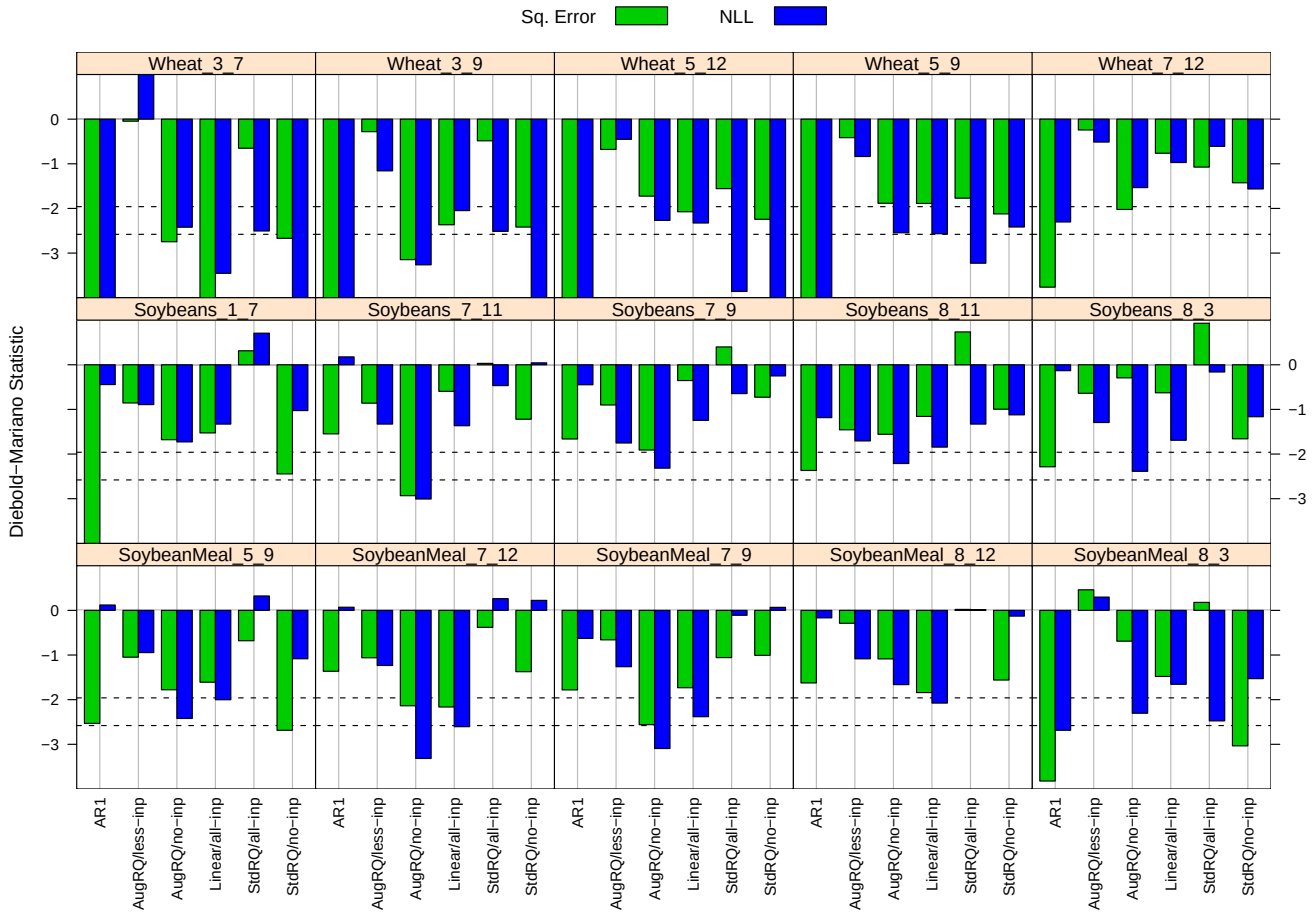
$$\widehat{IR}(t_1, t_2) = \frac{\mathbb{E}[\tilde{p}_{t_2} - \tilde{p}_{t_1} | \mathcal{I}_{t_0}]}{\sqrt{\mathrm{Var}[\tilde{p}_{t_2} - \tilde{p}_{t_1} | \mathcal{I}_{t_0}]}}, \tag{6.26}$$

where $\mathrm{Var}[\tilde{p}_{t_2} - \tilde{p}_{t_1} | \mathcal{I}_{t_0}]$ can be computed as $\mathrm{Var}[\tilde{p}_{t_1} | \mathcal{I}_{t_0}] + \mathrm{Var}[\tilde{p}_{t_2} | \mathcal{I}_{t_0}] - 2\,\mathrm{Cov}[\tilde{p}_{t_1}, \tilde{p}_{t_2} | \mathcal{I}_{t_0}]$. The Gaussian process model yields a forecast of the undiscounted quantities (*cf.* eq. (6.6)) $\mathrm{Var}[p_{t_1} | \mathcal{I}_{t_0}]$, $\mathrm{Var}[p_{t_2} | \mathcal{I}_{t_0}]$ and $\mathrm{Cov}[p_{t_1}, p_{t_2} | \mathcal{I}_{t_0}]$, and discounting can then be applied as noted above. The trade decision is made in one of two ways, depending on whether a position has already been opened:

*An *information ratio* is defined as the average return of a portfolio in excess of a benchmark, divided by the standard deviation of the excess return distribution; see (Grinold and Kahn 2000) for more details.

Table 6.1. *Forecast performance difference between* ***AugRQ/all-inp*** *and all other models, for the three spreads studied. For both the Squared Error and NLL criteria, the value of the cross-correlation-corrected statistic is listed (CCC) along with its p-value under the null hypothesis. A negative CCC statistic indicates that* ***AugRQ/all-inp*** *beats the other model on average.*

	SoybeanMeal 5–9				**SoybeanMeal 7–12**				**SoybeanMeal 7–9**				**SoybeanMeal 8–12**				**SoybeanMeal 8–3**			
	Sq. Error		NLL		Sq. Error		NLL		Sq. Error		NLL		Sq. Error		NLL		Sq. Error		NLL	
	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p
AugRQ/less-inp	−1.05	0.294	−0.94	0.344	−1.06	0.288	−1.23	0.217	−0.66	0.509	−1.26	0.207	−0.28	0.772	−1.08	0.277	0.46	0.644	0.29	0.765
AugRQ/no-inp	−1.78	0.075	−2.42	0.015	−2.13	0.032	−3.31	10^{-3}	−2.56	0.010	−3.09	0.001	−1.08	0.276	−1.66	0.096	−0.68	0.490	−2.30	0.021
Linear/all-inp	−1.61	0.107	−2.00	0.045	−2.16	0.030	−2.60	0.009	−1.73	0.083	−2.38	0.017	−1.84	0.065	−2.07	0.037	−1.48	0.138	−1.65	0.097
$AR(1)$	−2.53	0.011	0.12	0.904	−1.36	0.172	0.07	0.943	−1.78	0.074	−0.62	0.529	−1.62	0.104	−0.16	0.866	−3.82	10^{-4}	−2.68	0.007
StdRQ/all-inp	−0.68	0.496	0.323	0.747	−0.38	0.703	0.26	0.793	−1.06	0.290	−0.11	0.913	0.02	0.983	0.01	0.988	0.18	0.858	−2.48	0.013
StdRQ/no-inp	−2.69	0.007	−1.08	0.278	−1.37	0.169	0.22	0.822	−1.00	0.313	0.06	0.945	−1.56	0.118	−0.12	0.897	−3.03	0.002	−1.52	0.126
	Soybeans 1–7				**Soybeans 7–11**				**Soybeans 7–9**				**Soybeans 8–11**				**Soybeans 8–3**			
	Sq. Error		NLL		Sq. Error		NLL		Sq. Error		NLL		Sq. Error		NLL		Sq. Error		NLL	
	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p
AugRQ/less-inp	−0.86	0.392	−0.89	0.372	−0.86	0.389	−1.32	0.183	−0.90	0.367	−1.75	0.079	−1.45	0.145	−1.70	0.087	−0.63	0.522	−1.29	0.195
AugRQ/no-inp	−1.68	0.093	−1.72	0.083	−2.93	0.003	−3.00	0.002	−1.91	0.056	−2.31	0.020	−1.55	0.119	−2.21	0.026	−0.29	0.769	−2.38	0.017
Linear/all-inp	−1.53	0.126	−1.32	0.183	−0.59	0.551	−1.36	0.172	−0.35	0.723	−1.24	0.213	−1.15	0.247	−1.84	0.065	−0.62	0.530	−1.69	0.090
$AR(1)$	−4.25	10^{-5}	−0.44	0.657	−1.54	0.121	0.17	0.859	−1.66	0.096	−0.44	0.654	−2.36	0.017	−1.18	0.236	−2.28	0.022	−0.13	0.894
StdRQ/all-inp	0.31	0.754	0.71	0.479	0.03	0.975	−0.47	0.641	0.40	0.689	−0.65	0.518	0.74	0.462	−1.33	0.184	0.93	0.352	−0.16	0.871
StdRQ/no-inp	−2.44	0.014	−1.02	0.304	−1.21	0.222	0.04	0.965	−0.72	0.467	−0.24	0.802	−0.99	0.319	−1.12	0.261	−1.65	0.097	−1.16	0.244
	Wheat 3–7				**Wheat 3–9**				**Wheat 5–12**				**Wheat 5–9**				**Wheat 7–12**			
	Sq. Error		NLL		Sq. Error		NLL		Sq. Error		NLL		Sq. Error		NLL		Sq. Error		NLL	
	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p	CCC	p
AugRQ/less-inp	−0.05	0.963	1.05	0.291	−0.28	0.778	−1.15	0.246	−0.67	0.497	−0.45	0.651	−0.41	0.678	−0.83	0.402	−0.24	0.808	−0.51	0.607
AugRQ/no-inp	−2.75	0.006	−2.42	0.015	−3.14	0.001	−3.26	0.001	−1.72	0.084	−2.26	0.023	−1.88	0.059	−2.54	0.010	−2.02	0.043	−1.53	0.125
Linear/all-inp	−4.20	10^{-5}	−3.45	10^{-4}	−2.36	0.017	−2.04	0.040	−2.07	0.038	−2.32	0.019	−1.88	0.058	−2.56	0.010	−0.76	0.444	−0.97	0.331
$AR(1)$	−6.50	10^{-9}	−6.07	10^{-9}	−5.95	10^{-9}	−4.38	10^{-5}	−4.88	10^{-6}	−5.55	10^{-8}	−4.55	10^{-6}	−4.55	10^{-6}	−3.76	10^{-4}	−2.30	0.021
StdRQ/all-inp	−0.65	0.514	−2.51	0.012	−0.49	0.628	−2.51	0.012	−1.56	0.125	−3.86	10^{-4}	−1.77	0.077	−3.23	0.001	−1.07	0.283	−0.61	0.543
StdRQ/no-inp	−2.67	0.007	−9.35	0.000	−2.41	0.015	−5.50	10^{-8}	−2.24	0.024	−4.36	10^{-5}	−2.12	0.033	−2.41	0.015	−1.42	0.153	−1.56	0.117



▲ **Figure 6.9.** *Overview of the Cross-Correlation-Corrected Diebold-Mariano test results for the grain spreads. The 1% and 5% thresholds for statistical significance (two-tailed test) are indicated by the dashed horizontal lines.*

1. When making a decision at time t_0, if a position has **not yet been entered** for the spread in a given trading year, eq. (6.26) is maximized with respect to unconstrained $t_1, t_2 \geq t_0$. An illustration of this criterion is given in Figure 6.10, which corresponds to the first decision made when trading the spread shown in Figure 6.7.

2. In contrast, if a position **has already been opened**, eq. (6.26) is only maximized with respect to t_2, keeping t_1 fixed at t_0. This corresponds to revising the exit point of an existing position.

Simple additional filters are used to avoid entering marginal trades: we impose a trade duration of at least four days, a minimum forecast IR of 0.25 and a forecast standard deviation of the price sequence of at least 0.075. These thresholds have not been tuned extensively; they were used only to avoid trading on an approximately flat price forecast.*

We observe in passing that this framework is vastly different from a classical mean-variance portfolio optimization (Markowitz 1959), which would focus on *single-period* expected returns and variance thereof. The proposed framework for trading spreads, by relying on an explicit forecast of a *complete future trajectory*, is intuitive to practitioners, who can readily pass judgement about the accuracy of a forecast and the resulting suggested trades based on their own market view.

6.7 Financial Performance

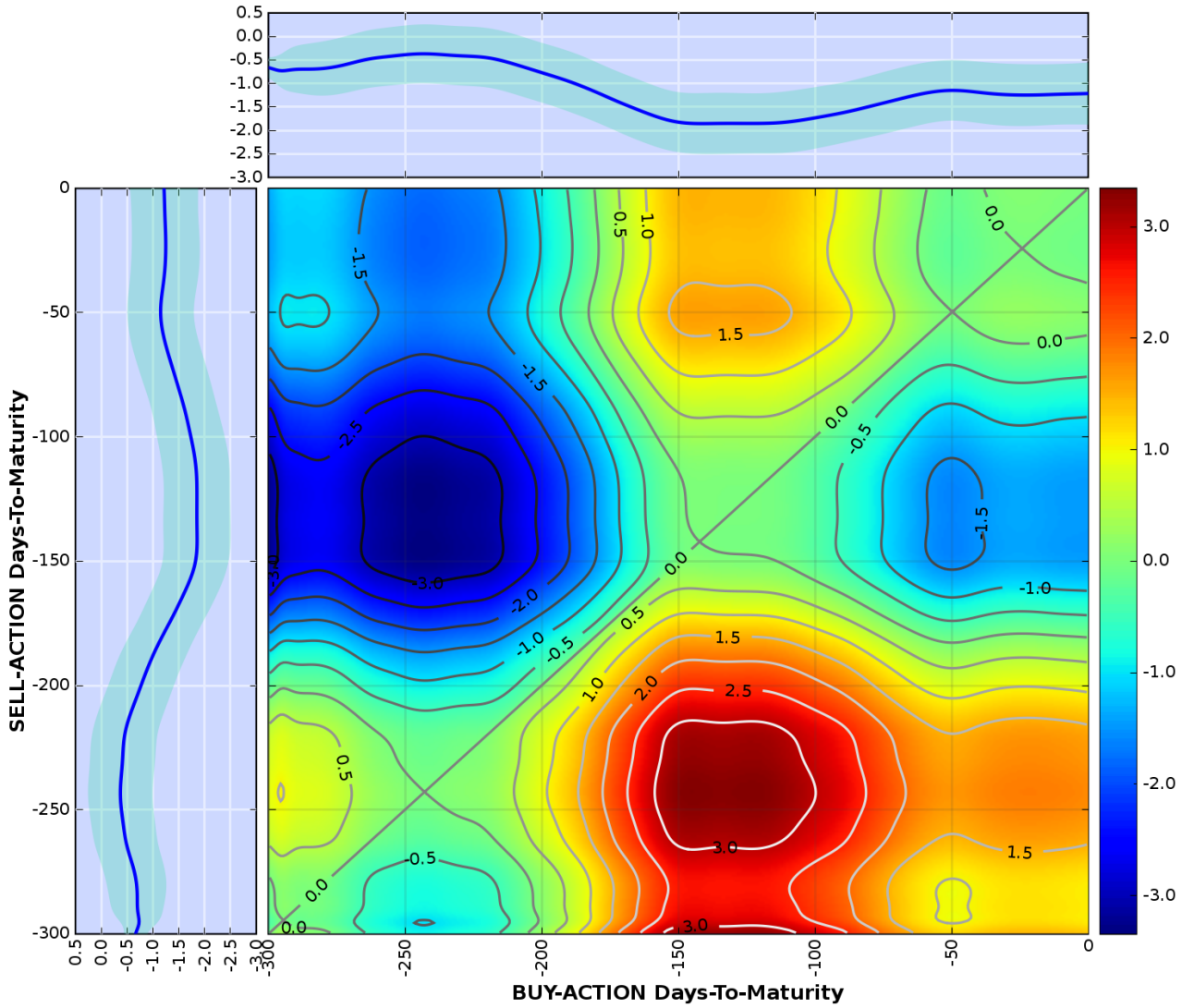
We applied these ideas to trading an equally-weighted portfolio of 30 spreads, selected among the following commodities: Cotton (2 spreads), Feeder Cattle (2), Gasoline (1), Lean Hogs (7), Live Cattle (1), Natural Gas (2), Soybean Meal (5), Soybeans (5), Wheat (5); the complete list of spreads traded appears in Table 6.2. The spreads were selected on the basis of their performance on the 1994–2002 period.

Our simulations were carried on the 1994–2007 period, using historical data (for Gaussian process training) dating back to 1989. Transaction costs were assumed to be 5 basis points per spread leg traded. Spreads were never traded later than 25 calendar days before maturity of the near leg. Relative returns are computed using as a notional amount half the total exposure incurred by both legs of the spread.† Moreover, since individual spreads trade only for a fraction of the year, returns are taken into consideration for a spread only when the spread is actually trading.

Financial performance results on the complete test period and two disjoint sub-periods (which correspond, until end-2002 to the model selection period, and after 2003 to a true out-of-sample evaluation) are shown in Tables 6.3 to 6.5. All results give returns in excess of the risk-free rate

†*This is a conservative assumption, since most exchanges impose considerably reduced margin requirements on recognized spreads.*

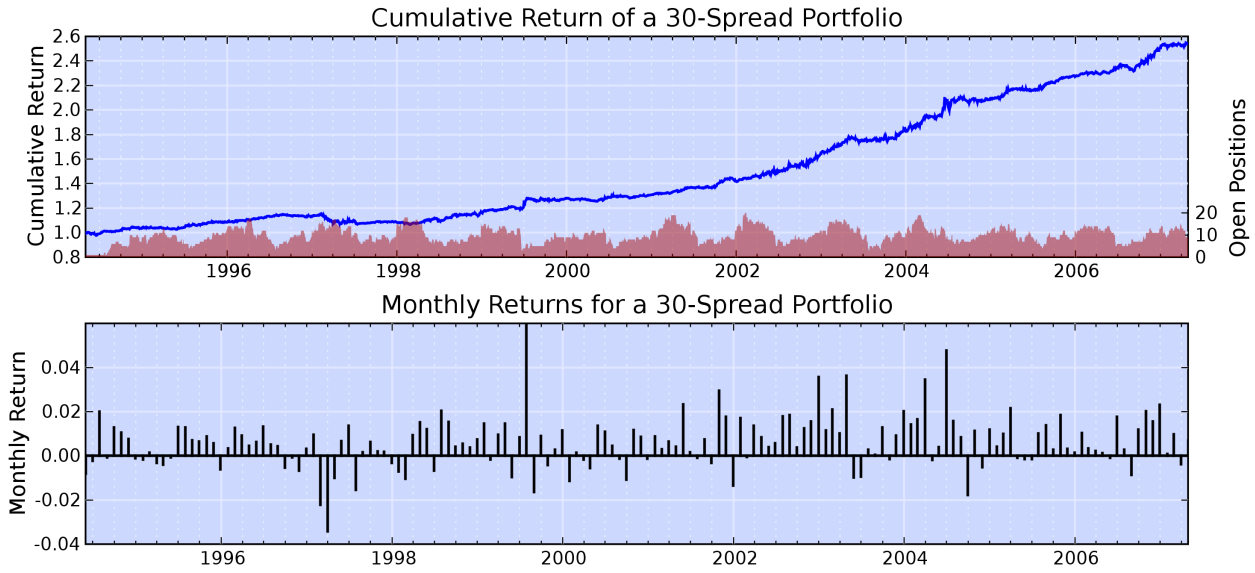
*The results are quite insensitive to the precise choice of these thresholds.



▲ **Figure 6.10.** *Computation of the Information Ratio between each potential entry and exit points, given the spread trajectory forecast. The x-axis represents the potential "buy" times, and the y-axis the potential "sell" times.*

Table 6.2. *List of spreads used to construct the tested portfolio.*

Commodity	**Maturities *short–long***
Cotton	10–12, 10–5
FeederCattle	11–3, 8–10
Gasoline	1–5
LeanHogs	12–2, 2–4, 2–6, 4–6, 7–12, 8–10, 8–12
LiveCattle	2–8
NaturalGas	6–9, 6–12
SoybeanMeal	5–9, 7–9, 7–12, 8–3, 8–12
Soybeans	1–7, 7–11, 7–9, 8–3, 8–11
Wheat	3–7, 3–9, 5–9, 5–12, 7–12



▲ **Figure 6.11.** ***Top Panel:*** *cumulative excess return after transaction costs of a portfolio of 30 spreads traded according to the maximum information-ratio criterion; the bottom part plots the number of positions open at a time (right axis).* ***Bottom Panel:*** *monthly portfolio relative excess returns; we observe the significant positive skewness in the distribution.*

(since margin deposits earn interest in a futures trading account). In all sub-periods, but particularly since 2003, the portfolio exhibits a very favorable risk-return profile, including positive skewness and acceptable excess kurtosis.*

Year-by-year returns, for the complete portfolio as well as sub-portfolios formed by the spreads on a single underlying commodity, are given in Table 6.6. Furthermore, the monthly-return correlation matrix among the sub-portfolios formed by same-commodity spreads appears is Table 6.7.

A plot of cumulative returns, number of open positions and monthly returns appears in Figure 6.11.

For comparison purposes, the financial performance of the $AR(1)$ and **Linear/all-inp** models on the 30-spread portfolio appear in Tables 6.8 and 6.9. Echoing the forecasting results of §6.5/p. 268, these two models significantly lag the **AugRQ/all-inp** model on key measures (in particular, the information ratio), on all subperiods.

*By way of comparison, over the period 1 Jan. 1994–30 Apr. 2007, the S&P 500 index has an information ratio of approximately 0.37 against the U.S. three-month treasury bills.

Table 6.3. *Performance on the 1 Jan. 1994–30 April 2007 period. All returns are expressed in excess of the risk-free rate. The information ratio statistics are annualized. Skewness and excess kurtosis are on the monthly return distributions. Drawdown duration is expressed in calendar days. The model shows good performance for moderate risk.*

	Portfolio	Cotton	F. Cattle	Gasoline	Lean Hogs
Annualized Return	7.4%	1.6%	1.8%	2.5%	5.2%
Avg Annual Return	7.3%	3.3%	2.6%	6.7%	5.6%
Avg Annual Stddev	4.1%	4.8%	4.0%	6.8%	8.3%
Annual Inf Ratio	1.77	0.68	0.65	0.99	0.67
Avg Monthly Return	0.6%	0.3%	0.2%	0.6%	0.5%
Avg Monthly Stddev	1.2%	1.4%	1.1%	2.0%	2.4%
Skewness	0.68	0.47	−0.07	0.13	0.32
Excess Kurtosis	3.40	2.63	4.04	0.66	1.11
Best Month	6.0%	5.4%	4.0%	6.1%	9.3%
Worst Month	−3.4%	−4.0%	−4.9%	−4.7%	−5.9%
Fraction Months Up	71%	56%	50%	67%	58%
Max. Drawdown	−7.7%	−6.8%	−7.0%	−6.5%	−12.9%
Drawdown Duration	653	705	774	1040	137
Drawdown From	1997/02	1997/05	1994/08	1996/11	1998/09
Drawdown Until	1998/11	1999/04	1996/09	1999/09	1999/01
	L. Cattle	Nat. Gas	SB Meal	Soybeans	Wheat
Annualized Return	1.5%	2.4%	2.8%	3.3%	7.0%
Avg Annual Return	3.8%	4.7%	4.3%	4.1%	8.6%
Avg Annual Stddev	7.6%	7.8%	8.8%	6.6%	6.5%
Annual Inf Ratio	0.50	0.60	0.49	0.61	1.33
Avg Monthly Return	0.3%	0.4%	0.4%	0.3%	0.7%
Avg Monthly Stddev	2.2%	2.2%	2.5%	1.9%	1.9%
Skewness	−0.35	0.74	2.23	−0.14	1.02
Excess Kurtosis	2.50	2.01	11.73	5.39	5.31
Best Month	5.7%	9.4%	14.1%	8.5%	9.5%
Worst Month	−8.1%	−5.2%	−6.9%	−7.7%	−6.5%
Fraction Months Up	58%	53%	53%	56%	71%
Max. Drawdown	−13.3%	−14.4%	−25.1%	−16.3%	−13.7%
Drawdown Duration	> 1286	1528	2346	1714	82
Drawdown From	2003/10	2001/02	1996/11	1996/12	2002/10
Drawdown Until	None	2005/04	2003/05	2001/09	2002/12

	Portfolio	Cotton	F. Cattle	Gasoline	Lean Hogs
Annualized Return	5.9%	2.0%	0.7%	1.6%	5.0%
Avg Annual Return	5.9%	4.3%	1.1%	4.5%	5.4%
Avg Annual Stddev	4.0%	5.3%	4.1%	7.0%	8.5%
Annual Inf Ratio	1.45	0.80	0.27	0.65	0.64
Avg Monthly Return	0.5%	0.4%	0.1%	0.4%	0.5%
Avg Monthly Stddev	1.2%	1.5%	1.2%	2.0%	2.4%
Skewness	0.65	0.42	−0.35	0.10	0.59
Excess Kurtosis	4.60	2.20	4.83	0.77	1.60
Best Month	6.0%	5.4%	4.0%	6.1%	9.3%
Worst Month	−3.4%	−4.0%	−4.9%	−4.7%	−5.4%
Fraction Months Up	67%	59%	47%	65%	58%
Max. Drawdown	−7.7%	−6.8%	−7.0%	−6.5%	−12.9%
Drawdown Duration	653	705	774	1040	137
Drawdown From	1997/02	1997/05	1994/08	1996/11	1998/09
Drawdown Until	1998/11	1999/04	1996/09	1999/09	1999/01
	L. Cattle	Nat. Gas	SB Meal	Soybeans	Wheat
Annualized Return	2.7%	0.5%	−0, 2%	2.3%	5.6%
Avg Annual Return	6.2%	1.2%	−0, 1%	2.8%	7.2%
Avg Annual Stddev	6.6%	8.2%	6, 5%	5.5%	6.5%
Annual Inf Ratio	0.94	0.14	−0, 01	0.52	1.11
Avg Monthly Return	0.5%	0.1%	0, 0%	0.2%	0.6%
Avg Monthly Stddev	1.9%	2.4%	1, 9%	1.6%	1.9%
Skewness	0.64	1.13	−0, 70	−1.19	0.91
Excess Kurtosis	0.41	3.02	3, 38	6.97	6.93
Best Month	5.7%	9.4%	5, 7%	5.2%	9.5%
Worst Month	−2.6%	−5.2%	−6, 9%	−7.7%	−6.5%
Fraction Months Up	62%	46%	51%	57%	68%
Maximum Drawdown	−8.2%	−11.7%	−25, 1%	−16.3%	−13.7%
Drawdown Duration	485	> 686	> 2225	1714	82
Drawdown From	1997/08	2001/02	1996/11	1996/12	2002/10
Drawdown Until	1998/12	None	None	2001/09	2002/12

Table 6.4. *Performance on the 1 January 1994–31 December 2002 period. All returns are expressed in excess of the risk-free rate. The information ratio statistics are annualized. Skewness and excess kurtosis are on the monthly return distributions. Drawdown duration is expressed in calendar days.*

Table 6.5. *Performance on the 1 January 2003–30 April 2007 period. All returns are expressed in excess of the risk-free rate. The information ratio statistics are annualized. Skewness and excess kurtosis are on the monthly return distributions. Drawdown duration is expressed in calendar days.*

	Portfolio	Cotton	F. Cattle	Gasoline	Lean Hogs
Annualized Return	10.5%	0.6%	4.1%	4.4%	5.8%
Avg Annual Return	10.1%	1.2%	5.4%	10.9%	6.0%
Avg Annual Stddev	4.1%	3.6%	3.6%	6.3%	8.1%
Annual Inf Ratio	2.44	0.34	1.49	1.73	0.73
Avg Monthly Return	0.8%	0.1%	0.4%	0.9%	0.5%
Avg Monthly Stddev	1.2%	1.0%	1.0%	1.8%	2.3%
Skewness	0.76	0.11	0.89	0.35	−0.28
Excess Kurtosis	1.26	0.07	0.24	−0.30	−0.15
Best Month	4.8%	2.7%	3.3%	5.1%	5.0%
Worst Month	−1.8%	−2.0%	−1.3%	−2.5%	−5.9%
Fraction Months Up	77%	50%	56%	71%	58%
Maximum Drawdown	−4.0%	−5.8%	−4.1%	−4.4%	−11.1%
Drawdown Duration	23	309	373	297	355
Drawdown From	2004/06	2003/08	2004/08	2003/09	2004/03
Drawdown Until	2004/07	2004/06	2005/08	2004/07	2005/03
	L. Cattle	Nat. Gas	SB Meal	Soybeans	Wheat
Annualized Return	−0.9%	6.3%	9.2%	5.5%	9.8%
Avg Annual Return	−1.8%	11.0%	12.0%	6.4%	11.3%
Avg Annual Stddev	9.5%	6.8%	11.6%	8.4%	6.4%
Annual Inf Ratio	−0.19	1.62	1.04	0.76	1.77
Avg Monthly Return	−0.2%	0.9%	1.0%	0.5%	0.9%
Avg Monthly Stddev	2.7%	2.0%	3.3%	2.4%	1.8%
Skewness	−0.86	−0.09	2.55	0.30	1.25
Excess Kurtosis	1.66	−0.10	7.18	2.73	1.67
Best Month	5.3%	4.7%	14.1%	8.5%	6.6%
Worst Month	−8.1%	−3.6%	−3.9%	−7.3%	−2.1%
Fraction Months Up	50%	66%	58%	55%	77%
Maximum Drawdown	−13.3%	−6.9%	−13.4%	−13.0%	−9.3%
Drawdown Duration	> 1286	112	16	334	69
Drawdown From	2003/10	2003/01	2004/06	2003/04	2003/10
Drawdown Until	None	2003/04	2004/07	2004/03	2003/12

Table 6.6. *Yearly returns of the entire portfolio and of sub-portfolios formed by spreads on the same underlying commodity. Year 2007 includes data until April 30 (the return reported is not annualized).*

	Portfolio	Cotton	F.Cattle	Gasoline	LeanHogs	L.Cattle	N.Gas	SB Meal	Soybeans	Wheat
1994	3.9%	0.0%	0.1%	0.0%	−0.6%	7.7%	−0.2%	−0.6%	−1.3%	8.6%
1995	4.0%	5.8%	1.5%	0.0%	4.0%	2.9%	−4.5%	1.9%	−1.4%	−0.8%
1996	5.0%	3.6%	−2.2%	−1.5%	−2.3%	1.5%	3.5%	3.8%	8.1%	0.6%
1997	−4.3%	0.1%	0.5%	2.8%	3.5%	−1.0%	−1.6%	−19.6%	−12.3%	14.5%
1998	7.1%	1.2%	2.4%	−0.5%	4.2%	5.5%	6.8%	−4.5%	2.7%	5.5%
1999	10.1%	0.6%	2.0%	4.6%	16.1%	2.5%	7.1%	4.7%	0.0%	4.1%
2000	1.9%	1.5%	0.4%	−1.4%	−1.6%	−4.8%	−1.2%	9.1%	4.7%	0.7%
2001	8.7%	2.7%	2.2%	10.8%	8.8%	3.3%	−5.0%	−0.9%	8.5%	7.0%
2002	16.2%	2.5%	−0.8%	0.0%	13.1%	6.4%	0.2%	7.7%	13.2%	9.9%
2003	10.8%	2.5%	8.2%	1.7%	4.4%	−2.8%	1.0%	3.4%	−1.4%	18.2%
2004	14.7%	−1.5%	−0.8%	7.8%	0.0%	0.7%	3.5%	40.7%	24.5%	4.1%
2005	8.5%	3.2%	5.0%	6.9%	7.8%	1.8%	5.4%	5.4%	2.5%	4.4%
2006	10.3%	−1.8%	3.6%	2.8%	13.0%	−1.1%	17.7%	−4.1%	0.0%	12.8%
2007	1.4%	0.3%	1.8%	0.0%	0.3%	−2.3%	0.5%	−0.8%	0.1%	3.5%

Table 6.7. *Correlation matrix of monthly returns among sub-portfolios formed by spreads on the same underlying commodity, on the 1994–2007 period.*

	Cotton	F.Cattle	Gasoline	LeanHogs	L.Cattle	Nat.Gas	SB Meal	Soybeans	Wheat
Cotton	—	0.06	−0.11	−0.04	−0.02	0.02	0.11	0.07	−0.09
FeederCattle	0.06	—	0.12	−0.05	−0.04	−0.05	−0.02	−0.09	0.00
Gasoline	−0.11	0.12	—	0.01	0.21	−0.06	0.04	−0.04	−0.02
LeanHogs	−0.04	−0.05	0.01	—	−0.14	−0.02	0.00	−0.02	0.01
LiveCattle	−0.02	−0.04	0.21	−0.14	—	−0.03	−0.01	−0.03	0.08
NaturalGas	0.02	−0.05	−0.06	−0.02	−0.03	—	−0.02	0.07	−0.13
SoybeanMeal	0.11	−0.02	0.04	0.00	−0.01	−0.02	—	0.40	−0.02
Soybeans	0.07	−0.09	−0.04	−0.02	−0.03	0.07	0.40	—	−0.04
Wheat	−0.09	0.00	−0.02	0.01	0.08	−0.13	−0.02	−0.04	—

Table 6.8. *Financial performance of the AR(1) model on the 30-spread portfolio, evaluated over the full period from 1 January 1994–30 April 2007, and on two disjoint subperiods. The AR(1) stronly lags the* ***AugRQ/all-inp*** *Gaussian process model (compare to Tables 6.3, 6.4 and 6.5).*

	Full Period	1994/01–2002/12	2003/01–2007/04
Annualized Return	1.0%	0.9%	1.5%
Avg Annual Return	1.3%	1.1%	1.8%
Avg Annual Stddev	7.0%	6.7%	7.8%
Annual Inf Ratio	0.18	0.17	0.23
Avg Monthly Return	0.1%	0.1%	0.2%
Avg Monthly Stddev	2.0%	1.9%	2.2%
Skewness	0.55	0.68	0.37
Excess Kurtosis	2.18	2.38	1.62
Best Month	8.4%	8.4%	6.5%
Worst Month	−5.6%	−5.2%	−5.6%
Fraction Months Up	54.2%	56.3%	50%
Maximum Drawdown	−18.5%	−18.5%	−13.8%
Drawdown Duration	1456	1456	192
Drawdown From	1995/07	1995/07	2005/11
Drawdown Until	1999/07	1999/07	—

Annual Returns (results for 2007 are until April 30th)

1994	−1.2%	2001	−1.8%
1995	0.7%	2002	4.3%
1996	2.1%	2003	5.4%
1997	−11.4%	2004	4.7%
1998	−2.0%	2005	5.8%
1999	21.9%	2006	−6.7%
2000	−2.4%	2007	−2.3%

	Full Period	1994/01–2002/12	2003/01–2007/04
Annualized Return	4.9%	5.2%	4.2%
Avg Annual Return	5.0%	5.3%	4.3%
Avg Annual Stddev	4.9%	3.9%	6.4%
Annual Inf Ratio	1.02	1.33	0.67
Avg Monthly Return	0.4%	0.4%	0.4%
Avg Monthly Stddev	1.4%	1.1%	1.8%
Skewness	−0.26	0.72	−0.65
Excess Kurtosis	3.61	1.33	2.57
Best Month	4.8%	4.2%	4.8%
Worst Month	−5.9%	−2.2%	−5.9%
Fraction Months Up	68%	69%	65%
Maximum Drawdown	−8.3%	−5.1%	−8.3%
Drawdown Duration	967	622	967
Drawdown From	2004/05	1996/03	2004/05
Drawdown Until	2007/01	1997/12	2007/01

Table 6.9. *Financial performance of the* ***Linear/all-inp*** *model on the 30-spread portfolio, evaluated over the full period from 1 January 1994–30 April 2007, and on two disjoint subperiods. This model exhibits an intermediate performance between the $AR(1)$ and the* ***AugRQ/all-inp*** *models (compare to Tables 6.3, 6.4 and 6.5).*

Annual Returns (results for 2007 are until April 30th)

1994	5.0%	2001	5.4%
1995	3.9%	2002	15.5%
1996	−2.1%	2003	6.5%
1997	4.5%	2004	3.1%
1998	6.5%	2005	2.9%
1999	2.9%	2006	4.1%
2000	5.3%	2007	1.7%

6.8 Discussion

This chapter introduced a flexible functional representation of time series, capable of making long-term forecasts from progressively-revealed information sets and of handling multiple irregularly-sampled series as training examples. We demonstrated the approach on a challenging commodity spread trading application, making use of a Gaussian process' ability to compute a complete covariance matrix between several test outputs.

An obvious limitation of the proposed approach comes from the computational complexity of Gaussian processes: whereas parameter estimation in an $AR(k)$ model requires time $O(Tk^3)$ (where T is the number of time-steps in the training set), the exact solution for Gaussian processes require time $O(T^3)$. The augmented functional representation proposed in this paper compounds the problem since it increases the training set size by at least a constant factor (depending on how subsampling is performed, as discussed in §6.3.4/p. 260). Future work includes making more systematic use of approximation methods for Gaussian processes (see Quiñonero-Candela and Rasmussen (2005) for a survey). The specific usage pattern of the Gaussian process may guide the approximation: in particular, since we know in advance the test inputs, the problem is intrinsically one of *transduction* (Vapnik 1998), and the Bayesian Committee Machine (Tresp 2000) could prove beneficial.

6.9 Appendix: Analysis of Forecasting Errors for the Wheat 3–7 Spread

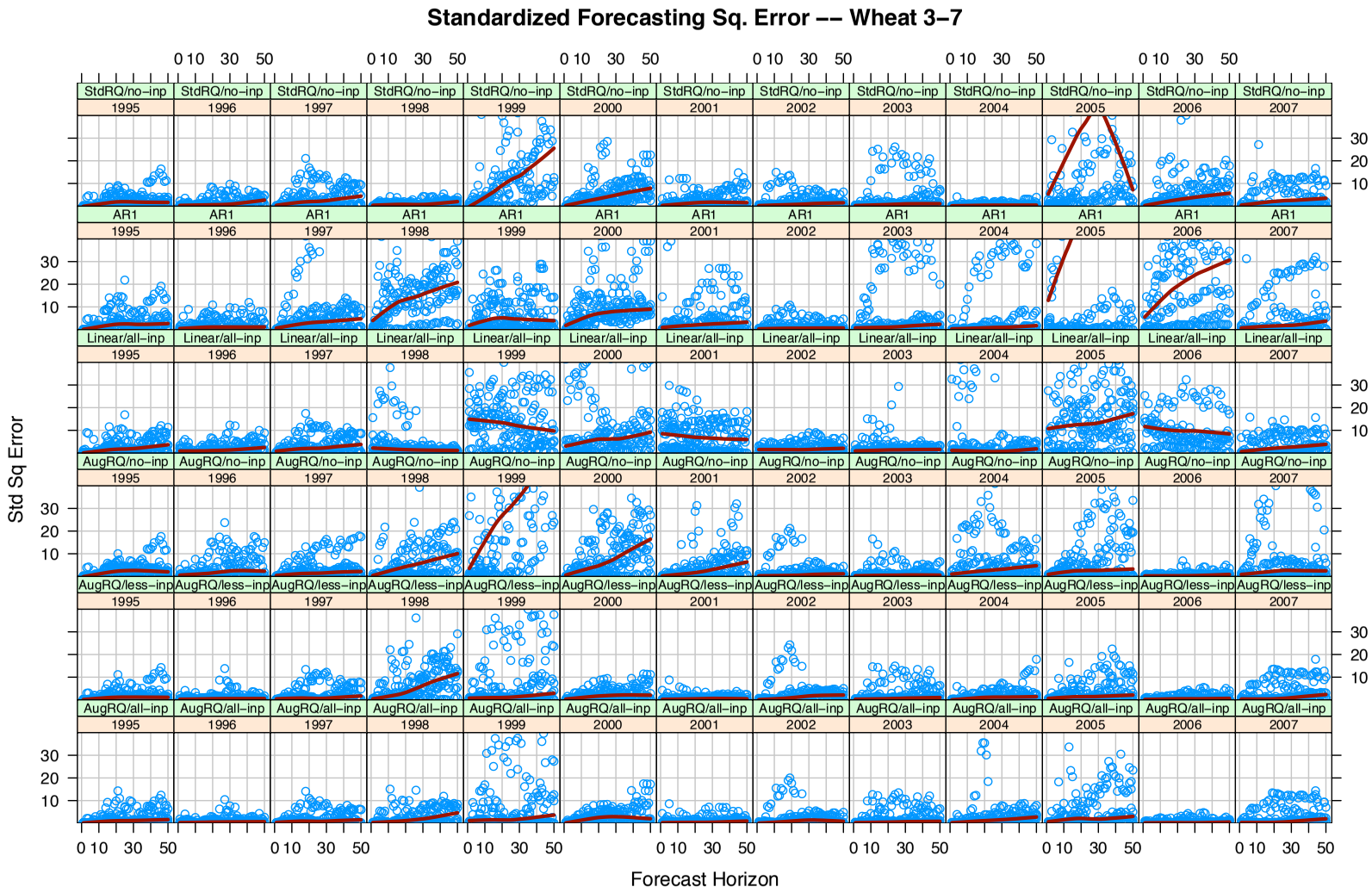
The following pages illustrate with more detail the forecasting performance of the models compared in §6.5/p. 268, for every year of the period 1994–2007 (until Apr. 30, for 2007), for the March–July Wheat spread.

We focus on the results (both for the standardized squared error and standardized negative log likelihood) *as a function of the forecast horizon* (in calendar days). Figures 6.12 and 6.13 give absolute performance figures for all models being compared and all years. The solid red line is the result of a local smoothing of the error and is useful to identify trends.

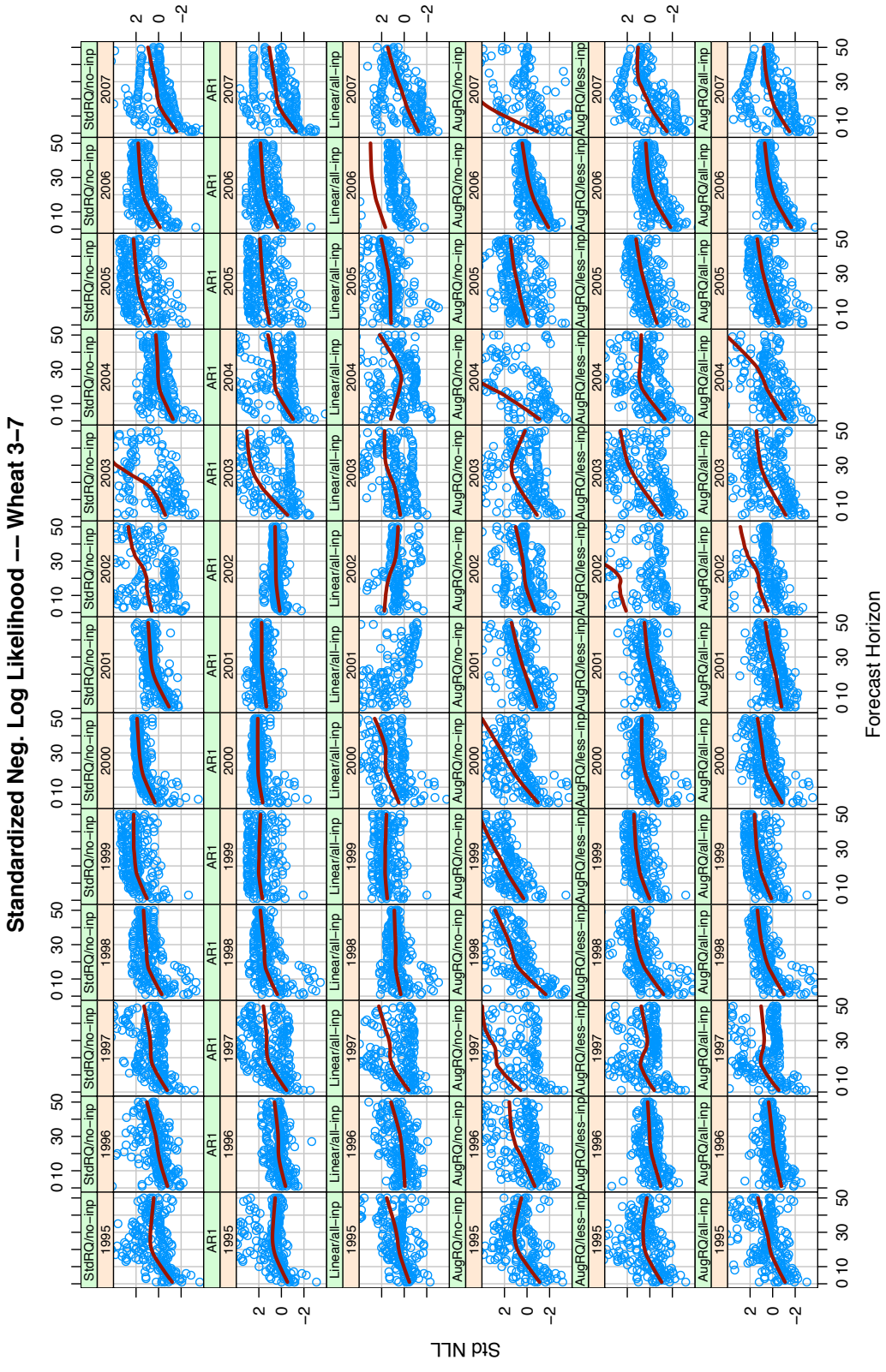
As a general rule, we note that performance degrades as the forecast horizon increases, as can be inferred from the upward-sloping trend line. Also, the augmented-representation Gaussian process with all variables, **AugRQ/all-inp** (the "reference model"), appears to be systematically as good or better than the other models.

This intuition is confirmed by Figures 6.14 to 6.23, which compare, in a pairwise fashion, **AugRQ/all-inp** against every other model. Performance measures (and the smoothed trend line) below the zero line indicate that **AugRQ/all-inp** performs better than the alternative.

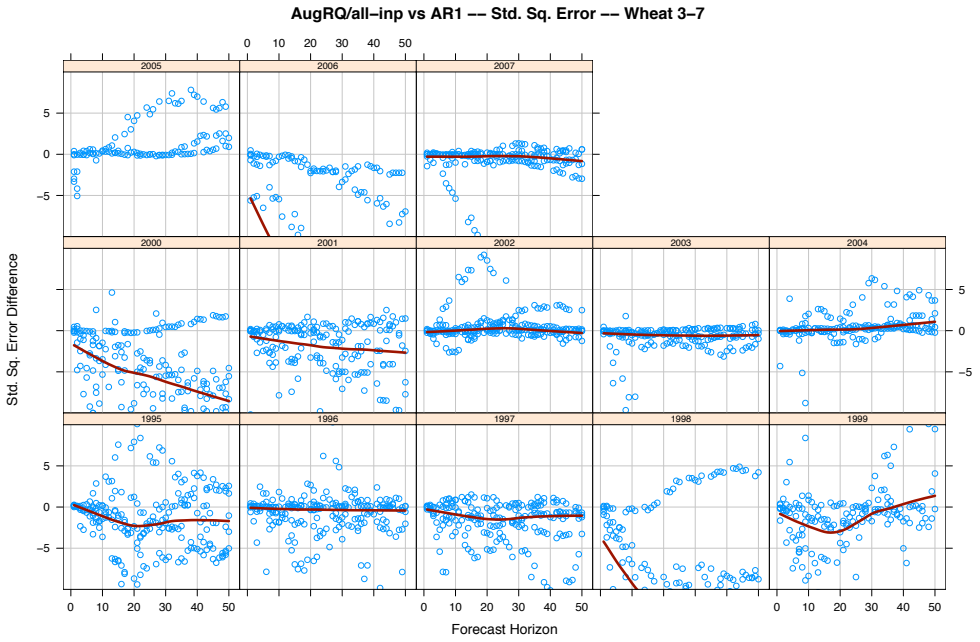
In many contexts, we see that this model beats the alternative (trend line below zero), but — just as significantly — its advantage increases with the forecast horizon, as witnessed by the *downward-sloping* trend line. In other words, even though the performance of both models generally tends to decrease with the horizon, **AugRQ/all-inp** generally holds its own better against the alternative and degrades less rapidly.



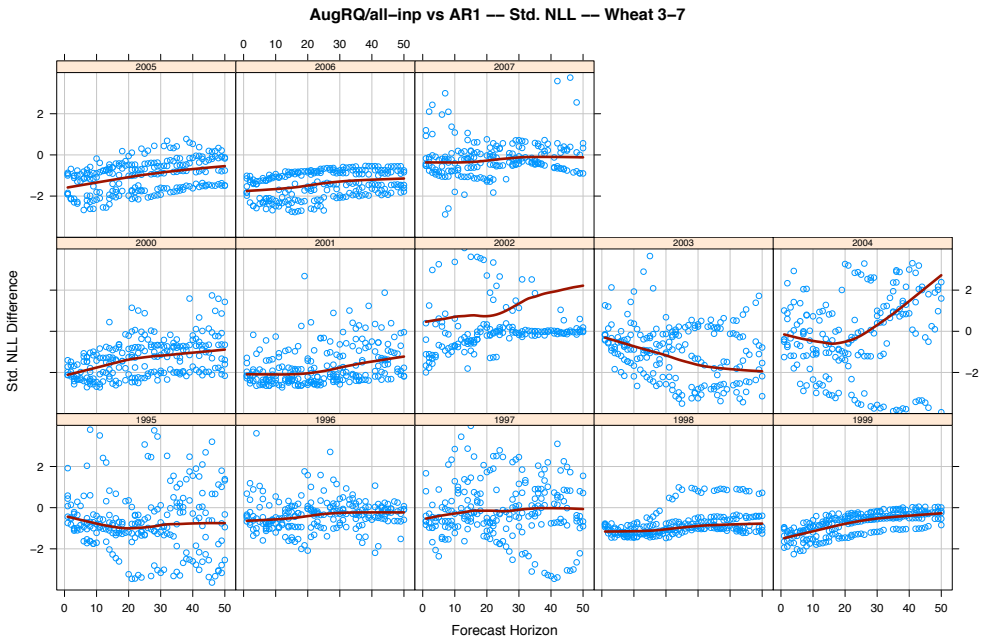
▲ **Figure 6.12.** *Standardized forecasting squared error as a function of the forecast horizon (calendar days) on the March–July Wheat spread, for all models considered in the main text and all years of the test period. We note that the mean forecast error tends to increase with the horizon.*



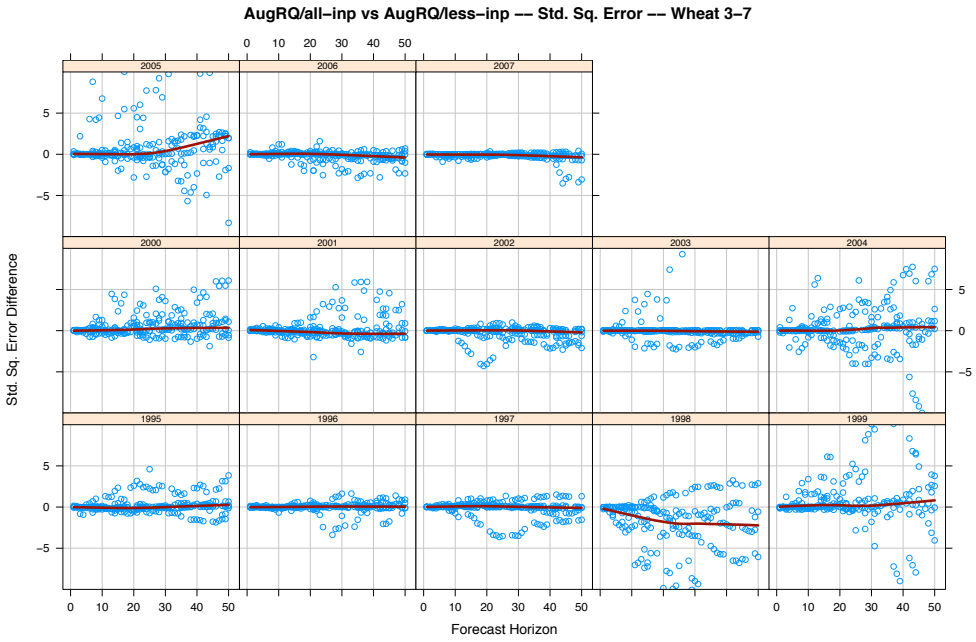
▲ **Figure 6.13.** *Standardized forecasting negative log likelihood as a function of the forecast horizon (calendar days) on the March–July Wheat spread, for all models considered in the main text and all years of the test period. We note that the mean forecast error tends to increase with the horizon.*



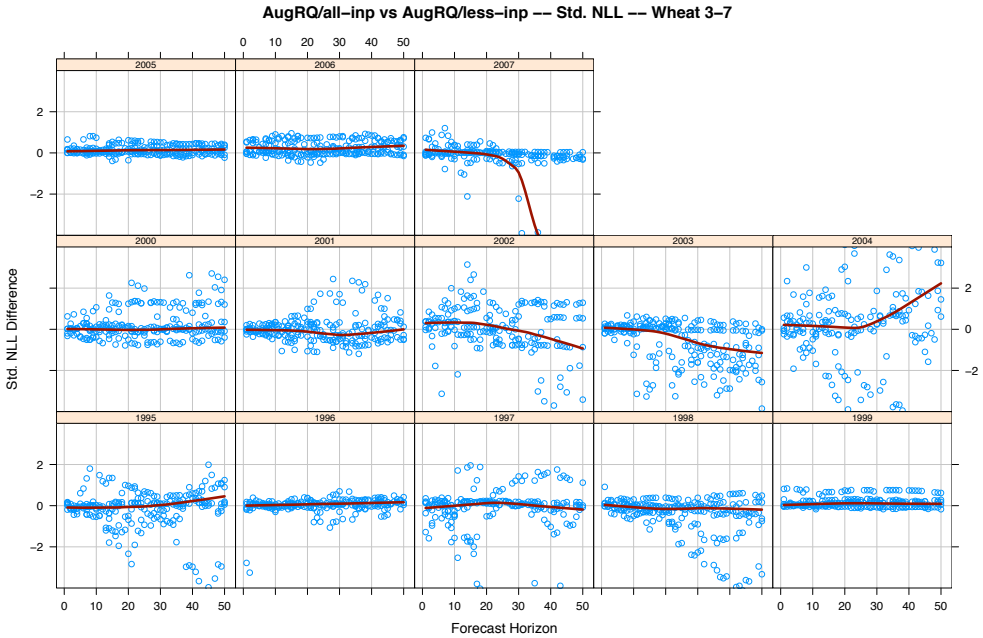
▶ **Figure 6.14.** *Squared error difference between* ***AugRQ/all-inp*** *and* ***AR1*** *as a function of the forecast horizon, for each year of the test period. A negative difference indicates an advantage for* ***AugRQ/all-inp****.*



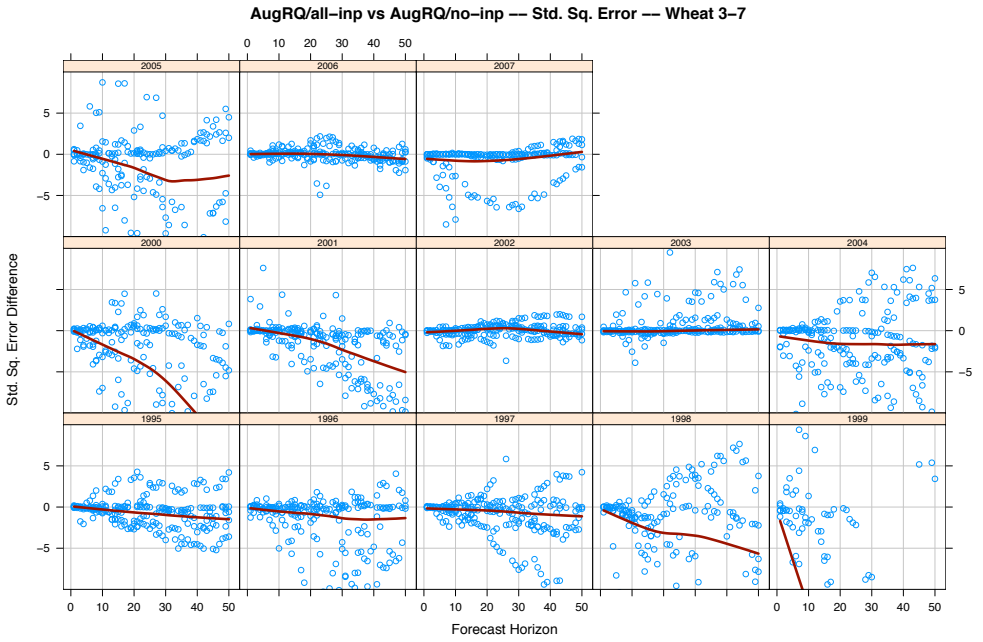
▶ **Figure 6.15.** *NLL difference between* ***AugRQ/all-inp*** *and* ***AR1*** *as a function of the forecast horizon, for each year of the test period. A negative difference indicates an advantage for* ***AugRQ/all-inp****.*



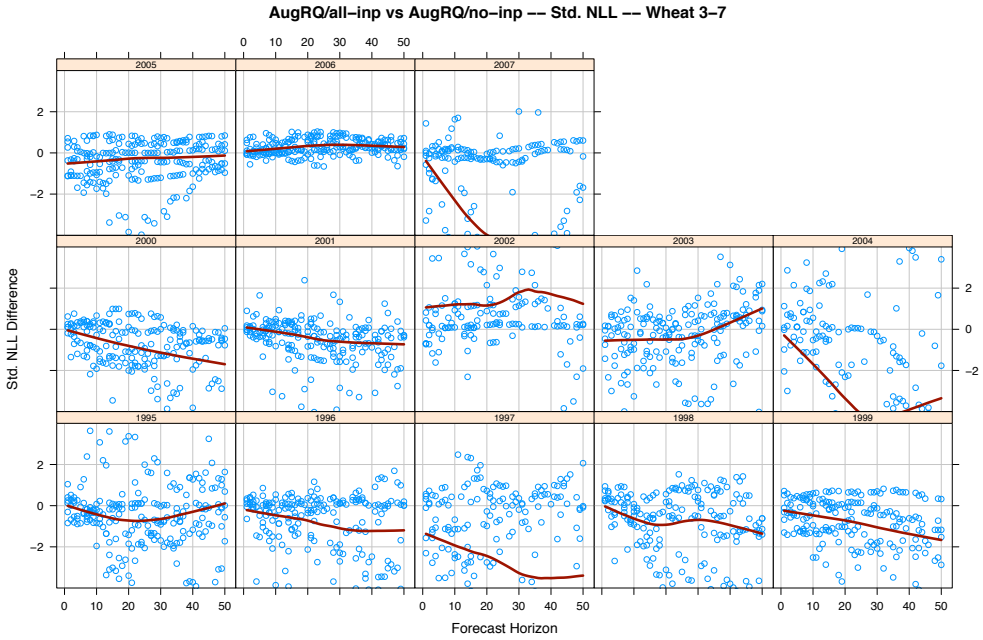
◀ **Figure 6.16.** *Squared error difference between* ***AugRQ/all-inp*** *and* ***AugRQ/less-inp*** *as a function of the forecast horizon, for each year of the test period. A negative difference indicates an advantage for* ***AugRQ/all-inp****.*



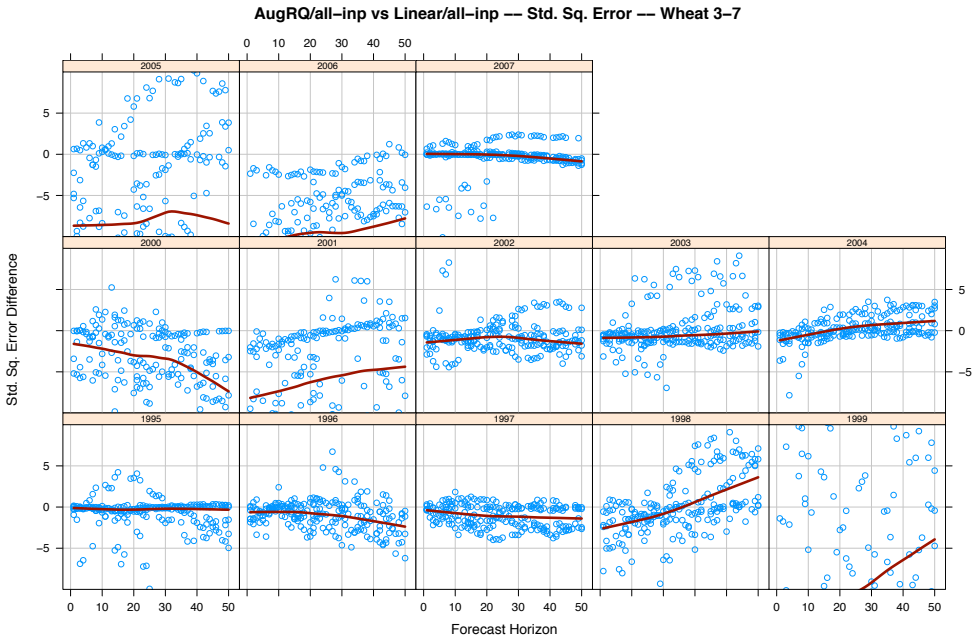
◀ **Figure 6.17.** *NLL difference between* ***AugRQ/all-inp*** *and* ***AugRQ/less-inp*** *as a function of the forecast horizon, for each year of the test period. A negative difference indicates an advantage for* ***AugRQ/all-inp****.*



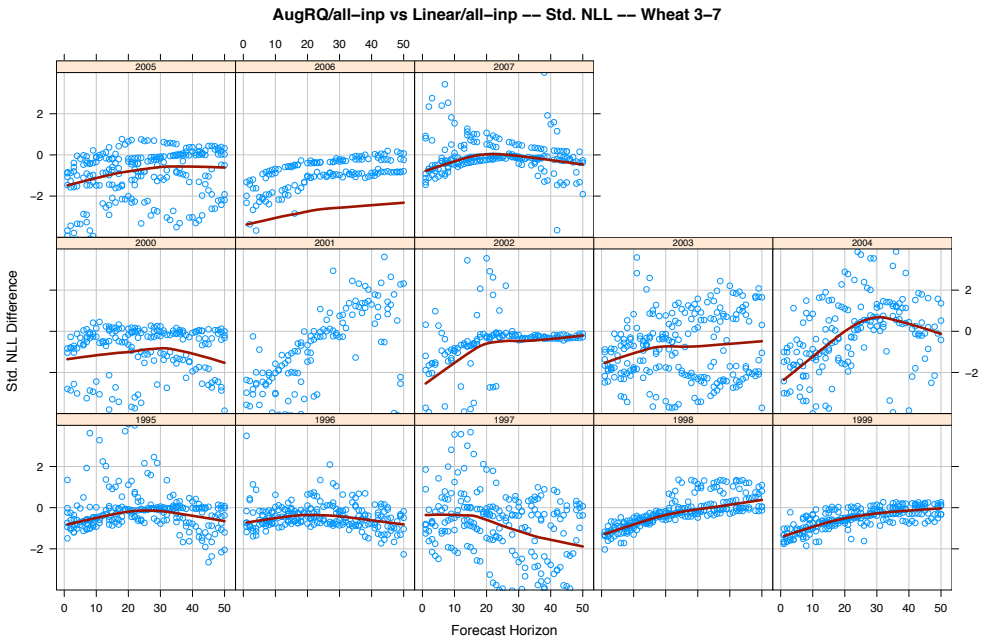
► **Figure 6.18.** *Squared error difference between* ***AugRQ/all-inp*** *and* ***AugRQ/no-inp*** *as a function of the forecast horizon, for each year of the test period. A negative difference indicates an advantage for* ***AugRQ/all-inp****.*



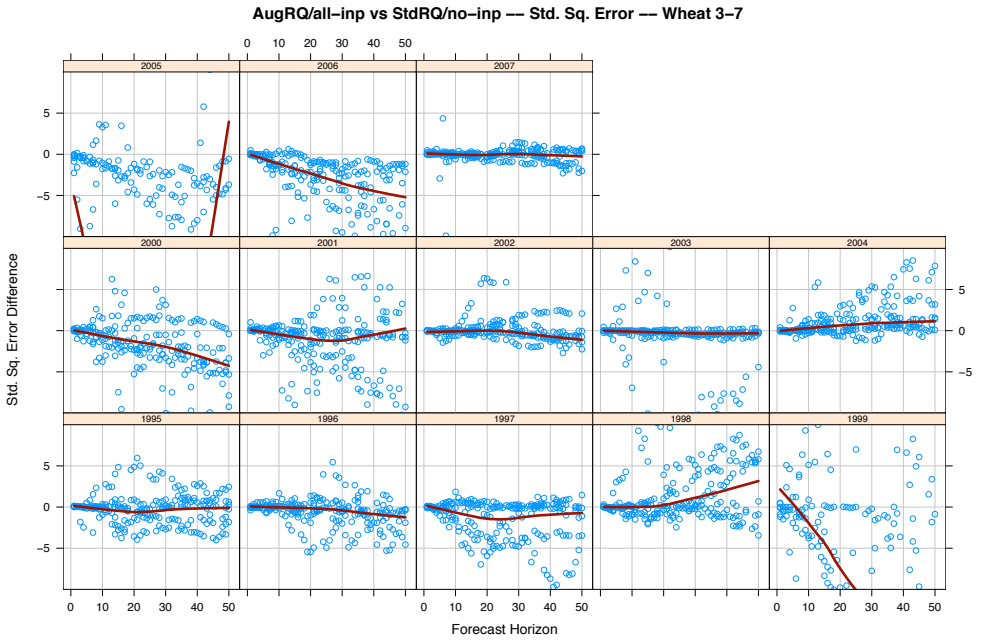
► **Figure 6.19.** *NLL difference between* ***AugRQ/all-inp*** *and* ***AugRQ/no-inp*** *as a function of the forecast horizon, for each year of the test period. A negative difference indicates an advantage for* ***AugRQ/all-inp****.*



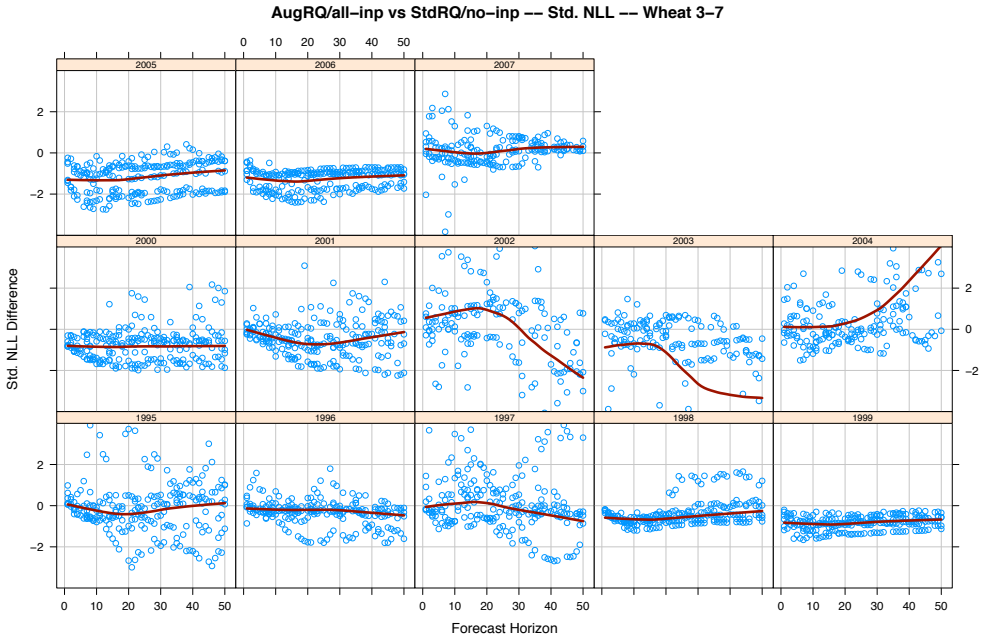
◀ **Figure 6.20.** *Squared error difference between* ***AugRQ/all-inp*** *and* ***Linear/all-inp*** *as a function of the forecast horizon, for each year of the test period. A negative difference indicates an advantage for* ***AugRQ/all-inp****.*



◀ **Figure 6.21.** *NLL difference between* ***AugRQ/all-inp*** *and* ***Linear/all-inp*** *as a function of the forecast horizon, for each year of the test period. A negative difference indicates an advantage for* ***AugRQ/all-inp****.*



▶ **Figure 6.22.** *Squared error difference between* ***AugRQ/all-inp*** *and* ***StdRQ/no-inp*** *as a function of the forecast horizon, for each year of the test period. A negative difference indicates an advantage for* ***AugRQ/all-inp****.*



▶ **Figure 6.23.** *NLL difference between* ***AugRQ/all-inp*** *and* ***StdRQ/no-inp*** *as a function of the forecast horizon, for each year of the test period. A negative difference indicates an advantage for* ***AugRQ/all-inp****.*

7 Conclusions and Summary of Contributions

It does not matter how slowly you go as long as you do not stop.

— *Confucius*

This thesis considered a number of approaches to make machine learning algorithms better suited to the sequential nature of financial portfolio management tasks. We summarize the progress made, and outline directions for future research.

7.1 Summary of Contributions

7.1.1 Chapter 2: Portfolio Choice

Although this chapter does not contain any original research, it is to our knowledge the first time where a comprehensive survey of the entire field of portfolio management (including both single-period and multiperiod formulations, as well as alternative methods) appears in an accessible form. There remains a considerable gap between the academic understanding of portfolio choice, which includes the subtle and elegant results of the Samuelson–Merton multiperiod formulation and the more recent martingale approach, with practitioners' implementations, which are mostly concerned with making single-period models robust.

Much remains to be accomplished to achieve a unified theory that deals as readily with estimation uncertainty and econometric issues, transaction costs, taxes and the changing investment opportunities that arise over long horizons, yet remains accessible and relevant to practitioners. We hope that a concise summary of the most significant elements can provide a step in this direction, and given the author's specialization in machine learning, an understandable entry point for computer scientists.

7.1.2 Chapter 4: Training Graphs of Learning Modules for Sequential Data

This chapter is of a more methodological nature and underscores the engineering challenges in composing graphs of learning modules to deal with sequential data. These issues have traditionally been neglected in the machine learning literature. We introduced a set of learning primitives that provide

a suitable abstraction for a large number of classical sequential learning algorithms, yet provide the necessary "hooks" to allow effective composition. We presented algorithms that efficiently handle the training-set updates that naturally arise within a sequential validation framework. Finally, we presented a detailed case study of a complex financial decision-making system making use of those learning primitives.

7.1.3 Chapter 5: K-Best Paths Methods for Portfolio Management

The main purpose of this chapter was to investigate the use of non-time-separable utility functions in portfolio management tasks, in particular the realized Sharpe ratio, which may be of more practical relevance than the utility functions considered in the standard theory.

We started by analyzing the suitability of using the classical Sharpe ratio criterion as an optimization objective (rather than a simple *ex post* comparison objective as it was originally designed to perform), and outlined its shortcomings for this new purpose. We proposed modifications to the criterion to regularize its behavior and introduce a preferred leverage scale. We illustrated the optimal policies that emerge under the Sharpe ratio criterion and showed that the action at time t depends on the realization of past returns. This is consistent with the observed behavior of institutional portfolio managers. Moreover, we showed that as soon as transaction costs are introduced, the Sharpe ratio can no longer be optimized by simple gradient-based algorithms.

To answer the problem of optimizing realized Sharpe ratios over realized asset price trajectories, we suggested using a method inspired from one used in speech recognition and consisting in extracting the K best paths under a tractable (time-separable) utility function (called the *source utility*), and rescoring under the utility function of interest (here, for instance, the Sharpe ratio, called the *target utility*). This approach is suitable if both the source and target utility functions are well matched.

We considered in detail two source utilities. First, the `MinCost` function corresponds to a maximum-profit criterion (subject to transaction costs). For this utility function, we established bounds on the value function which enable efficient pruning of the search graph, first by beam searching (state-space reduction) and efficient enumeration of the actions (action-space reduction). Together, and jointly with the use of realized trajectories, they attempt to tackle the three curses of dimensionality of dynamic programming (namely the sizes of the state, action and disturbance spaces; see the discussion in Powell (2007)). The second source utility is an incremental approximation to the Sharpe ratio, called `IncrSharpe`, and whose non-additivity results in a *noisy K-best paths extraction*. We established that the K-best paths approach is quite effective in optimizing realized Sharpe ratios in the presence of high transaction costs, whereas a gradient-based

algorithm awfully fails in this context.

As to making use of the results of the K-best paths search, we investigated a variant of a multi-layer neural network for performing ordinal regression. We proposed a parameterization of an ordinal regression neural network that is suitable for portfolio optimization and can be trained efficiently (called Financial Neural Network, FNNET); we also proposed a training objective that simultaneously takes into account an overall financial criterion as well as regularization terms that hint at good intermediate solutions and prevent overfitting. We show experimentally that a two-stage training procedure, where the network is first trained on a pure classification criterion, followed by a full financial objective, yields much better results than either criterion by itself.

Comparing the proposed FNNET to other standard regression and classification models across a number of markets, the proposed model is consistently among the best-performing ones. However, at this juncture, using the K-best paths for training the FNNET model (as part of the "hint" term in the objective function) does not improve performance over using more standard targets such as monthly or daily returns.

One of the messages to be taken from this chapter concerns the importance of regularization for ensuring good out-of-sample financial performance. On a very consistent basis, more constrained models perform much better, out of sample, than more flexible ones (within reason; namely, it is often possible to beat the naïve buy-and-hold, although not always statistically significantly). The form of regularization can be varied and creative: for instance, FNNET makes use of three different terms in its final objective function that can be interpreted as having a regularizing effect, without including the necessary prefitting stage which is another.

7.1.4 Chapter 6: Augmented Functional Time Series Representation and Forecasting with Gaussian Processes

In this last chapter, we attacked the time-series forecasting problem from a different angle than traditional methodologies such as ARIMA models.

We introduced an augmented functional representation of time-series, based on Gaussian processes, that allows variable-horizon forecasting and handling multiple seasonally-related time series. This is in sharp contrast to classical time-series models in which the forecasting horizon is fixed and implicit in the model.

We made use of the great flexibility afforded by Gaussian processes by making use of *covariance function engineering* to tailor the similarity structure to the seasonalities; this allows the model to exploit seasonality in its forecasts, but pay close attention to the specificities of the current series.

Regarding Gaussian process training, we exhibited an hyperparameter overfitting phenomenon in the two-stage fitting procedure of Gaussian processes. We also proposed an inverse softplus parameterization for positive

hyperparameters in Gaussian process covariance functions, which is found to yield considerably improved numerical stability.

In order to allow comparing the out-of-sample forecasting performance of several models evaluated by sequential validation, we introduced a correction for cross-covariance in the classical Diebold-Mariano test. The resulting CCC-DM test statistic is used to show that the proposed augmented functional representation generally performs significantly better out of sample, on both log-likelihood and squared error criteria, than simpler models.

We also introduced a trading criterion based on the predictive information ratio to turn the forecasts arising from the Gaussian process into trading decisions. Interestingly, this criterion makes use of the full joint predictive covariance matrix arising from a Gaussian process forecast, an attractive property of this model. Extensive experimentation with this criterion results in good financial performance results on a portfolio of 30 commodity spreads.

7.2 Outlook for Future Research

The current investigation leaves a number of stones unturned, and perhaps gives rise to as many questions as it answers.

In Chapter 5, we almost did not consider higher-dimensional problems and the results of allocating proper portfolios of assets. The attendant effects of diversification at longer horizons, and the form that hedging demand terms take when optimizing under nonstandard criteria should be investigated. Moreover, it appears of necessity to be able to train a controller that can at least use the current portfolio state as input, as well as specifically-sequential machine learning architectures, such as recurrent neural networks.

In Chapter 6, the results obtained so far have been quite constrained by the use of an exact solution to Gaussian processes, which restricts the size of the training set to the capacity of matrix inversion procedures; the use of sparse approximation techniques, which would allow training sets at least an order of magnitude larger, is a next logical step. This could help answer the question of the extent to which forecasting ability suffers from the subsampling that must be applied to training examples in order to use the augmented representation. On a different note, more complex noise dynamics should be incorporated into the model; although simple autoregressive noise structure can readily be incorporated via straightforward additions to the covariance function, more financially-relevant dynamics, such as stochastic volatility, are desirable. A fertile ground, which would allow seasonality-specific noise, might lie with input-dependent noise models, such as that of Goldberg, Williams, and Bishop (1998).

More generally, we can ask about the directions of machine learning applications in finance. All such endeavors must address two fundamental

properties of financial markets: (i) adequately handling the nonstationarities in the generating process, and (ii) accounting for the high noise levels in the data. Of the first property, two subtypes can be identified: gradual changes in the distribution and structural breaks. This first property—nonstationarities—suggests the appropriateness of considering *switching models*, perhaps multi-layered ones, to deal with nonstationarities differing in degree and in kind. The second property—noise—suggests the application of Bayesian methods to systematically transmit modeling and predictive uncertainty across all components of a forecasting system. In these respects, recent work on nonparametric Bayesian Markov switching dynamical systems in machine learning appears promising (Fox, Sudderth, Jordan, and Willsky 2009).

A final question regards the modeling process itself: although it should be a great leap of faith to assume that informational shocks would be incorporated linearly into the state variables driving underlying generating processes in economic systems and financial markets, such assumptions are routinely made in many models. On the contrary, it appears to many practitioners that information diffusion is somehow state-dependent: a given piece of news can have a great impact in some context, but be inconsequential in a different one. This could be considered akin to *occlusion effects* that arise in computer vision. The proper modeling of such *informational occlusion* remains an area of exciting investigation and one in which machine learning could bring fruitful contribution.

A Appendix

Little experience is sufficient to show that the traditional machinery of statistical processes is wholly unsuited to the needs of practical research.

— *Sir Ronald A. Fisher*

A.1 Minimization of a Quadratic Form Under Linear Equality Constraints

In §2.1/p. 9, we are faced with the problem of minimizing a quadratic form subject to linear equality constraints,

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{2}\,\mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w} \tag{A.1}$$

$$\text{subject to} \quad \mathbf{A}\mathbf{w} = \mathbf{b}, \tag{A.2}$$

where $\mathbf{w} \in \mathbb{R}^N, \boldsymbol{\Sigma} \in \mathbb{R}^{N\times N}, \mathbf{A} \in \mathbb{R}^{M\times N}, \mathbf{b} \in \mathbb{R}^M$. Let $\boldsymbol{\lambda} \in \mathbb{R}^M$ be a vector of Lagrange multipliers. We consider the Lagrangian function

$$\mathcal{L}(\mathbf{w}, \boldsymbol{\lambda}) = \frac{1}{2}\,\mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w} + \boldsymbol{\lambda}'(\mathbf{A}\mathbf{w} - \mathbf{b}). \tag{A.3}$$

The first-order conditions for optimality are obtained by differentiating (A.3) with respect to each variable and setting the resulting functions to zero,

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}'} = \boldsymbol{\Sigma}\,\mathbf{w} + \mathbf{A}'\boldsymbol{\lambda} = 0, \tag{A.4}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}'} = \mathbf{A}\mathbf{w} - \mathbf{b} \quad = 0. \tag{A.5}$$

This implies the following system of equation, which can be written as a partitioned matrix equation

$$\begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{A}' \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix}. \tag{A.6}$$

Assuming that the inverse of $\left(\begin{smallmatrix} \boldsymbol{\Sigma} & \mathbf{A}' \\ \mathbf{A} & \mathbf{0} \end{smallmatrix}\right)$ exists, the solution is given by

$$\begin{pmatrix} \mathbf{w} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\Sigma} & \mathbf{A}' \\ \mathbf{A} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ \mathbf{b} \end{pmatrix}. \tag{A.7}$$

The inverse of the partitioned matrix is obtained as (*cf.* Greene 2007)*

$$\begin{pmatrix} \mathbf{\Sigma} & \mathbf{A}' \\ \mathbf{A} & \mathbf{0} \end{pmatrix}^{-1} = \begin{pmatrix} \mathbf{\Sigma}^{-1}(\mathbf{I} + \mathbf{A}'\mathbf{F}\mathbf{A}\mathbf{\Sigma}^{-1}) & -\mathbf{\Sigma}^{-1}\mathbf{A}'\mathbf{F} \\ -\mathbf{F}\mathbf{A}\mathbf{\Sigma}^{-1} & \mathbf{F} \end{pmatrix}, \tag{A.8}$$

with $\mathbf{F} = -(\mathbf{A}\mathbf{\Sigma}^{-1}\mathbf{A}')^{-1}$. It can be shown that this inverse exists if $\mathbf{\Sigma}^{-1}$ exists and $\mathbf{A}$ is of full rank. Substituting in eq. (A.7), we have

$$\mathbf{w} = -\mathbf{\Sigma}^{-1}\mathbf{A}'\mathbf{F}\mathbf{b} \tag{A.9}$$

$$= \mathbf{\Sigma}^{-1}\mathbf{A}'(\mathbf{A}\mathbf{\Sigma}^{-1}\mathbf{A}')^{-1}\mathbf{b}. \tag{A.10}$$

A.2 Deriving the Tangency Portfolio

Consider the minimum-variance formulation of the mean-variance portfolio optimization problem (*cf.* §2.1.1/p. 10),

$$\min_{\mathbf{w}} \quad \frac{1}{2}\,\mathbf{w}'\mathbf{\Sigma}\,\mathbf{w} \tag{A.11}$$

$$\text{subject to} \quad \mathbf{w}'\boldsymbol{\mu} = \rho, \tag{A.12}$$

$$\mathbf{w}'\boldsymbol{\iota} = 1, \tag{A.13}$$

where $\boldsymbol{\mu}$ and $\mathbf{\Sigma}$ are respectively the vector of expected asset returns and covariance matrix between asset returns (assumed to be nonsingular), and ρ is a target portfolio return that remains unspecified. Let $\mathcal{W}_e$ be the *efficient frontier*, the set of all portfolios solving this problem (obtained by varying ρ). The tangency portfolio is the portfolio belonging to $\mathcal{W}_e$ having the largest return per unit of standard deviation,

$$\mathbf{w}^* = \arg\max_{\mathbf{w}\in\mathcal{W}_e} \frac{\mathbf{w}'\boldsymbol{\mu}}{\sqrt{\mathbf{w}'\mathbf{\Sigma}\,\mathbf{w}}}. \tag{A.14}$$

This portfolio is also seen as having a *maximal Sharpe Ratio* (Sharpe 1966, 1994).† The tangency portfolio is obtained in two steps: (i) characterization of the efficient frontier, and (ii) maximization of eq. (A.14).

*This can be verified by direct multiplication, i.e. $\left(\begin{smallmatrix} \mathbf{\Sigma} & \mathbf{A}' \\ \mathbf{A} & \mathbf{0} \end{smallmatrix}\right)^{-1}\left(\begin{smallmatrix} \mathbf{\Sigma} & \mathbf{A}' \\ \mathbf{A} & \mathbf{0} \end{smallmatrix}\right) = \mathbf{I}$.

†It should be noted that absent the sum-to-one constraint, pure maximization of the Sharpe Ratio is an ill-posed problem. To see this, consider scaling the positions of portfolio P by a positive constant γ. This yields Sharpe Ratio

$$SR_{\gamma P} = \frac{\mathbb{E}[\gamma R_P]}{\sqrt{\mathrm{Var}[\gamma R_P]}} = \frac{\gamma\mathbb{E}[R_P]}{\gamma\sqrt{\mathrm{Var}[R_P]}} = SR_P.$$

Hence in order to maximize the Sharpe Ratio, it is necessary to choose the scaling factor, which corresponds to establishing the target portfolio risk level. Alternatively, enforcing a sum-to-one constraint specifies a risk level as well. Sharpe Ratio maximization, despite occasional claims to the contrary, does not absolve one from specifying risk preferences.

A.2.1 Efficient Frontier

Portfolios on the efficient frontier are those that satisfy the following first-order conditions for optimality, obtained by incorporating constraints (A.12) and (A.13) into the objective through Lagrange multipliers and differentiating with respect to $\mathbf{w}$,

$$\boldsymbol{\Sigma}\,\mathbf{w} - \lambda_1\boldsymbol{\mu} - \lambda_2\boldsymbol{\iota} = 0,$$

yielding an optimal portfolio of the form

$$\mathbf{w}^* = \boldsymbol{\Sigma}^{-1}(\lambda_1\boldsymbol{\mu} + \lambda_2\boldsymbol{\iota}). \tag{A.15}$$

The Lagrange multipliers are obtained by substituting this solution back into the constraints, and must jointly satisfy

$$\lambda_1\boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \lambda_2\boldsymbol{\mu}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\iota} = \rho \tag{A.16}$$

and

$$\lambda_1\boldsymbol{\iota}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \lambda_2\boldsymbol{\iota}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\iota} = 1. \tag{A.17}$$

A.2.2 Maximization of the Sharpe Ratio

The first-order conditions for the solution of eq. (A.14) are obtained by differentiating with respect to $\mathbf{w}$, yielding

$$\frac{\boldsymbol{\mu}}{\sqrt{\mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w}}} - \frac{\mathbf{w}'\boldsymbol{\mu}}{(\mathbf{w}'\boldsymbol{\Sigma}\mathbf{w})^{3/2}}\boldsymbol{\Sigma}\mathbf{w} = 0,$$

which simplifies to

$$(\mathbf{w}'\boldsymbol{\mu})\boldsymbol{\Sigma}\mathbf{w} = (\mathbf{w}'\boldsymbol{\Sigma}\,\mathbf{w})\boldsymbol{\mu}$$

and implies

$$\mathbf{w}^* = \frac{\mathbf{w}^{*\prime}\boldsymbol{\Sigma}\mathbf{w}^*}{\mathbf{w}^{*\prime}\boldsymbol{\mu}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}. \tag{A.18}$$

Since the tangency portfolio belongs to the efficient frontier, it must have the general form given by eq. (A.15). Comparing with eq. (A.18), this is only possible if $\lambda_2 = 0$. Substituting into eq. (A.17), we obtain

$$\lambda_1 = \frac{1}{\boldsymbol{\iota}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}},$$

which finally yields the desired **functional form for the tangency portfolio**,

$$\mathbf{w}^* = \frac{\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}{\boldsymbol{\iota}'\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}}.$$

This can be interpreted as follows: the numerator assigns weight to "virtual assets" (formed by decorrelated linear combinations of the original assets)

proportionally of their individual expected-return/variance ratio, and transforms them back into the space of original assets. The denominator acts as a normalization term (sum of the elements) to ensure that the elements of $\mathbf{w}^*$ sum to one.

If there is a risk-free asset, the same result obtains if we instead consider $\boldsymbol{\mu}$ to be the vector of expected *excess* asset returns. This amounts to shifting down the efficient frontier by a constant equal to the risk-free rate.

A.3 Itô's Lemma

Itô's celebrated lemma (Itô 1951) is central to any study of continuous-time financial models involving stochastic differential equations. It shows how to express the differential of a (sufficiently smooth) function of a random process. Standard textbooks on stochastic calculus cover this material, such as Shreve (2005a, 2005b). Its use in the context of continuous-time portfolio optimization appears in §2.2.2/p. 44.

A.3.1 Wiener Processes

Define the stochastic process $Z(t)$ as

$$Z(t+h) = Z(t) + y(t)\sqrt{h},$$

where $y(t)$ is a process of IID standard normal variables (i.e. zero-mean and unit variance) and $h > 0$. It can be observed that $Z(t+h) \sim N(z(t), h)$. In the limit as $h \to 0$, the increment $Z(t+h) - Z(t)$ follows a standard Wiener process and is defined as

$$\mathrm{d}Z \triangleq y(t)\sqrt{\mathrm{d}t}.$$

The Wiener process serves as a building block of more complex processes in continuous-time finance. For example, a simple process modeling the evolution of stock prices (sometimes termed a *geometric Brownian motion*) is

$$\frac{\mathrm{d}P_t}{P_t} = \mu(t)\mathrm{d}t + \sigma(t)\mathrm{d}Z.$$

The interpretation of this equation is that the *relative* change in the price P_t of the stock is given by the sum of a deterministic return $\mu(t)$ and a stochastic return proportional to $\sigma(t)$. Both $\mu(t)$ and $\sigma(t)$ are here deterministic functions of time. The process P_t is continuous but nowhere differentiable.

A.3.2 One-Dimensional Case

Consider the Markov random process $X(t)$ specified as

$$\mathrm{d}X(t) = \mu(X(t), t)\mathrm{d}t + \sigma(X(t), t)\mathrm{d}Z(t),$$

where $\mathrm{d}Z(t)$ is a standard Wiener process, denoted

$$\mathrm{d}Z(t) = u(t)\sqrt{\mathrm{d}t}, \qquad u(t) \overset{\text{iid}}{\sim} N(0,1).$$

The functions $\mu(X,t)$ and $\sigma(X,t)$ are, respectively, deterministic functions giving the *drift rate* and *volatility* of the process.

Lemma 12 *Let $f : \mathbb{R} \times [0,\infty] \mapsto \mathbb{R}$ a square-integrable function. Then the random process $f(X(t),t)$ is given by the differential*

$$\mathrm{d}f = \frac{\partial f(X,t)}{\partial X}\mathrm{d}X + \frac{\partial f(X,t)}{\partial t}\mathrm{d}t + \frac{1}{2}\frac{\partial^2 f(X,t)}{\partial X^2}(\mathrm{d}X)^2 \tag{A.19}$$

where the product of differentials is given by the multiplication rules

$$(\mathrm{d}Z)^2 = \mathrm{d}t, \qquad \mathrm{d}Z\,\mathrm{d}t = 0, \qquad (\mathrm{d}t)^2 = 0.$$

A.3.3 Multi-Dimensional Case

The generalization to the multi-dimensional case obtains readily. We shall consider the case where the dimensionality of the process $\mathbf{X}(t)$ is the same as the underlying sources of uncertainty. Let $\mathbf{X}(t) \in \mathbb{R}^N$ be specified as

$$\mathrm{d}\mathbf{X}(t) = \boldsymbol{\mu}(\mathbf{X}(t),t)\mathrm{d}t + \boldsymbol{\sigma}(\mathbf{X},t)\mathrm{d}\mathbf{Z}(t),$$

where $\mathrm{d}\mathbf{Z}(t)$ is a correlated Wiener process

$$\mathrm{d}\mathbf{Z}(t) = \mathbf{u}(t)\sqrt{\mathrm{d}t}, \qquad \mathbf{u}(t) \overset{\text{iid}}{\sim} N(0,\boldsymbol{\Gamma})$$

with $(\boldsymbol{\Gamma})_{i,j} \equiv \rho_{i,j}$ is the *correlation* between variables Z_i and Z_j; we assume them to have unit variance. The functions $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ are suitably generalized to be N-vector-valued.

Lemma 13 *Let $f : \mathbb{R}^N \times [0,\infty] \mapsto \mathbb{R}$ a square-integrable function. Then the random process $f(\mathbf{X}(t),t)$ is given by the differential*

$$\mathrm{d}f = \sum_{i=1}^{N}\frac{\partial f(\mathbf{X},t)}{\partial \mathbf{X}_i}\mathrm{d}\mathbf{X}_i + \frac{\partial f(\mathbf{X},t)}{\partial t}\mathrm{d}t + \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\frac{\partial^2 f(\mathbf{X},t)}{\partial \mathbf{X}_i\,\partial \mathbf{X}_j}\mathrm{d}\mathbf{X}_i\,\mathrm{d}\mathbf{X}_j \tag{A.20}$$

where the product of differentials is given by the multiplication rules

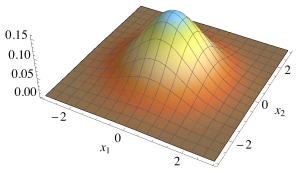
$$(\mathrm{d}\mathbf{Z}_i\,\mathrm{d}\mathbf{Z}_j) = \rho_{i,j}\,\mathrm{d}t, \qquad \mathrm{d}Z_i\,\mathrm{d}t = 0, \qquad (\mathrm{d}t)^2 = 0.$$

A.4 Inference for the Normal Distribution

We briefly derive the expressions arising from conditioning and marginalizing a multivariate normal distribution; these are useful in the analysis of Gaussian processes in Chapter 6.

Plot of normal density

$p_{\mathcal{N}}\left(\left(\begin{smallmatrix} x_1 \\ x_2 \end{smallmatrix}\right) \middle| \left(\begin{smallmatrix} 0 \\ 0 \end{smallmatrix}\right), \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)\right)$



The multivariate normal distribution (also known as the Gaussian distribution) with mean $\boldsymbol{\mu} \in \mathbb{R}^D$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{D\times D}$, denoted $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, has a density function given by

$$p_{\mathcal{N}}(\mathbf{x} \,|\, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{D/2}\,|\boldsymbol{\Sigma}|^{1/2}} \exp\Big(-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu})'\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})\Big), \tag{A.21}$$

where $|\boldsymbol{\Sigma}|$ denotes the determinant of the nonnegative definite matrix $\boldsymbol{\Sigma}$.

Consider a *partition* of the D variables into, without loss of generality the first M and remaining $N = D - M$ variables, and write

$$\boldsymbol{\mu} = \begin{pmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}.$$

It is useful to note that the determinant of a partitioned matrix can be written as

$$\begin{vmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{vmatrix} = |\boldsymbol{\Sigma}_{22}| \cdot \left|\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\right| = |\boldsymbol{\Sigma}_{11}| \cdot \left|\boldsymbol{\Sigma}_{22} - \boldsymbol{\Sigma}_{21}\boldsymbol{\Sigma}_{11}^{-1}\boldsymbol{\Sigma}_{12}\right|. \tag{A.22}$$

Denote the inverse of the covariance matrix, the *precision matrix*, by $\boldsymbol{\Lambda} \equiv \boldsymbol{\Sigma}^{-1}$. Analogously to the inverse of a partitioned matrix already seen in eq. (A.8), we can express the precision matrix as

$$\boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_{11} & \boldsymbol{\Lambda}_{12} \\ \boldsymbol{\Lambda}_{21} & \boldsymbol{\Lambda}_{22} \end{pmatrix} = \begin{pmatrix} \left(\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\right)^{-1} & -\boldsymbol{\Lambda}_{11}\mathbf{B} \\ -\mathbf{B}'\boldsymbol{\Lambda}_{11} & \boldsymbol{\Sigma}_{22}^{-1} + \mathbf{B}'\boldsymbol{\Lambda}_{11}\mathbf{B} \end{pmatrix} \tag{A.23}$$

where we have defined, for simplicity, $\mathbf{B} = \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}$. These equations can be verified by direct multiplication (i.e. $\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{-1} = I$).

A.4.1 Marginalization and Conditioning

Given a distribution over all variables $\mathbf{x} \equiv (\mathbf{x}_1', \mathbf{x}_2')'$, their joint density is written as the product of a conditional and a marginal,

$$p(\mathbf{x}_1, \mathbf{x}_2) = p(\mathbf{x}_1 \,|\, \mathbf{x}_2)\, p(\mathbf{x}_2).$$

We obtain the specific densities for each factor by substituting the precision matrix, as factored by eq. (A.23) into the joint density (A.21), making use of eq. (A.22) for the determinant and expanding terms,

$$\begin{aligned} p(\mathbf{x}_1, \mathbf{x}_2) = {} & \frac{1}{\left((2\pi)^{M/2}\left|\boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}\right|^{1/2}\right)\left((2\pi)^{N/2}|\boldsymbol{\Sigma}_{22}|^{1/2}\right)} \times \\ & \exp\Big(-\tfrac{1}{2}(\mathbf{x}_1-\boldsymbol{\mu}_1)'\boldsymbol{\Lambda}_{11}(\mathbf{x}_1-\boldsymbol{\mu}_1) \;-\tfrac{1}{2}(\mathbf{x}_1-\boldsymbol{\mu}_1)'\boldsymbol{\Lambda}_{12}(\mathbf{x}_2-\boldsymbol{\mu}_2) \\ & \quad -\tfrac{1}{2}(\mathbf{x}_2-\boldsymbol{\mu}_2)'\boldsymbol{\Lambda}_{21}(\mathbf{x}_1-\boldsymbol{\mu}_1) \;-\tfrac{1}{2}(\mathbf{x}_2-\boldsymbol{\mu}_2)'\boldsymbol{\Lambda}_{22}(\mathbf{x}_2-\boldsymbol{\mu}_2)\Big). \end{aligned} \tag{A.24}$$

Expanding $\mathbf{\Lambda}_{22}$ according to its definition, a first portion factors out to yield

$$p(\mathbf{x}_2) = \frac{1}{(2\pi)^{N/2}|\mathbf{\Sigma}_{22}|^{1/2}} \exp\Big(-\frac{1}{2}(\mathbf{x}_2-\boldsymbol{\mu}_2)'\mathbf{\Sigma}_{22}^{-1}(\mathbf{x}_2-\boldsymbol{\mu}_2)\Big) \tag{A.25}$$

$$= p_{\mathcal{N}}(\mathbf{x}_2 \,|\, \boldsymbol{\mu}_2, \mathbf{\Sigma}_{22}), \tag{A.26}$$

which establishes that the marginal density of a subset of the variables of a multivariate normal density is also normal, with the obvious mean and covariance matrix. Ignoring the $-\frac{1}{2}$ factors everywhere for now and expanding the remaining part of the last term, we have (the overbraces and underbraces will be used to indicate how terms are collected):

$$(\mathbf{x}_2-\boldsymbol{\mu}_2)'\mathbf{B}'\mathbf{\Lambda}_{11}\mathbf{B}(\mathbf{x}_2-\boldsymbol{\mu}_2) = \underbrace{\mathbf{x}_2'\mathbf{B}'\mathbf{\Lambda}_{11}\mathbf{B}\mathbf{x}_2} - \overbrace{\mathbf{x}_2'\mathbf{B}'\mathbf{\Lambda}_{11}\mathbf{B}\boldsymbol{\mu}_2} - \underbrace{\boldsymbol{\mu}_2'\mathbf{B}'\mathbf{\Lambda}_{11}\mathbf{B}\mathbf{x}_2} + \overbrace{\boldsymbol{\mu}_2'\mathbf{B}'\mathbf{\Lambda}_{11}\mathbf{B}\boldsymbol{\mu}_2}.$$

The middle terms (which, after transposition, are equal) yield

$$2(\mathbf{x}_1-\boldsymbol{\mu}_1)'\mathbf{\Lambda}_{12}(\mathbf{x}_2-\boldsymbol{\mu}_2) = -\underbrace{2\mathbf{x}_1'\mathbf{\Lambda}_{11}\mathbf{B}\mathbf{x}_2} + \overbrace{2\mathbf{x}_1'\mathbf{\Lambda}_{11}\mathbf{B}\boldsymbol{\mu}_2} + \underbrace{2\boldsymbol{\mu}_1'\mathbf{\Lambda}_{11}\mathbf{B}\mathbf{x}_2} - \overbrace{2\boldsymbol{\mu}_1'\mathbf{\Lambda}_{11}\mathbf{B}\boldsymbol{\mu}_2}.$$

Collecting and adding the $\underbrace{}$ and $\overbrace{}$ terms in the previous two equations yield, respectively,

$$\underbrace{} = \quad \big(\mathbf{B}(\mathbf{x}_2-\boldsymbol{\mu}_2) - 2(\mathbf{x}_1-\boldsymbol{\mu}_1)\big)'\mathbf{\Lambda}_{11}\mathbf{B}\mathbf{x}_2$$

$$\overbrace{} = -\big(\mathbf{B}(\mathbf{x}_2-\boldsymbol{\mu}_2) - 2(\mathbf{x}_1-\boldsymbol{\mu}_1)\big)'\mathbf{\Lambda}_{11}\mathbf{B}\boldsymbol{\mu}_2,$$

which, after addition and simplification, give

$$\big(\mathbf{B}(\mathbf{x}_2-\boldsymbol{\mu}_2) - 2(\mathbf{x}_1-\boldsymbol{\mu}_1)\big)'\mathbf{\Lambda}_{11}\mathbf{B}(\mathbf{x}-\boldsymbol{\mu}_2).$$

Define $\tilde{\boldsymbol{\mu}}_1 = \boldsymbol{\mu}_1 + \mathbf{B}(\mathbf{x}_2-\boldsymbol{\mu}_2)$. The last expression can be rewritten as

$$-(\mathbf{x}_1-\tilde{\boldsymbol{\mu}}_1)'\mathbf{\Lambda}_{11}(\tilde{\boldsymbol{\mu}}_1-\boldsymbol{\mu}_1) - (\mathbf{x}_1-\boldsymbol{\mu}_1)'\mathbf{\Lambda}_{11}(\tilde{\boldsymbol{\mu}}_1-\boldsymbol{\mu}_1).$$

The only missing part to the factorization is the very first term inside the exponential of eq. (A.24), $(\mathbf{x}_1-\boldsymbol{\mu}_1)'\mathbf{\Lambda}_{11}(\mathbf{x}_1-\boldsymbol{\mu}_1)$. Adding it to the previous expression, we obtain

$$\begin{aligned}
&\underbrace{(\mathbf{x}_1-\boldsymbol{\mu}_1)'\mathbf{\Lambda}_{11}(\mathbf{x}_1-\boldsymbol{\mu}_1)} - \overbrace{(\mathbf{x}_1-\tilde{\boldsymbol{\mu}}_1)'\mathbf{\Lambda}_{11}(\tilde{\boldsymbol{\mu}}_1-\boldsymbol{\mu}_1)} - \underbrace{(\mathbf{x}_1-\boldsymbol{\mu}_1)'\mathbf{\Lambda}_{11}(\tilde{\boldsymbol{\mu}}_1-\boldsymbol{\mu}_1)} \\
&\qquad = \quad (\mathbf{x}_1-\boldsymbol{\mu}_1)'\mathbf{\Lambda}_{11}(\mathbf{x}_1-\tilde{\boldsymbol{\mu}}_1) - (\tilde{\boldsymbol{\mu}}_1-\boldsymbol{\mu}_1)'\mathbf{\Lambda}_{11}(\mathbf{x}_1-\tilde{\boldsymbol{\mu}}_1) \\
&\qquad = \quad (\mathbf{x}_1-\tilde{\boldsymbol{\mu}}_1)'\mathbf{\Lambda}_{11}(\mathbf{x}_1-\tilde{\boldsymbol{\mu}}_1),
\end{aligned}$$

where the first equality arises, respectively, from simplifying the $\underbrace{}$ terms and exploiting the symmetry of the $\overbrace{}$ term, and the second equality comes

from a straightforward simplification. Bringing back the missing $-\frac{1}{2}$ factor, we see that all terms in eq. (A.24) are accounted for, which lets us write the conditional probability of $\mathbf{x}_1$ given $\mathbf{x}_2$ as

$$\begin{aligned} p(\mathbf{x}_1 \,|\, \mathbf{x}_2) &= \frac{1}{\left((2\pi)^{M/2}|\boldsymbol{\Lambda}_{11}|^{-1/2}\right)} \exp\Big(-\frac{1}{2}(\mathbf{x}_1 - \tilde{\boldsymbol{\mu}}_1)'\boldsymbol{\Lambda}_{11}(\mathbf{x}_1 - \tilde{\boldsymbol{\mu}}_1)\Big) \\ &= p_{\mathcal{N}}(\mathbf{x}_1 \,|\, \boldsymbol{\mu}_{1|2}, \boldsymbol{\Sigma}_{1|2}), \end{aligned} \tag{A.27}$$

with

$$\begin{aligned} \boldsymbol{\mu}_{1|2} &= \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\mathbf{x}_2 - \boldsymbol{\mu}_2) \\ \boldsymbol{\Sigma}_{1|2} &= \boldsymbol{\Lambda}_{11}^{-1} \quad = \quad \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \end{aligned}$$

thereby showing that the conditional distribution of $\mathbf{x}_1$ is normal with a "smaller" covariance matrix than $\boldsymbol{\Sigma}_{11}$ and a shifted mean reflecting the information gained by observing $\mathbf{x}_2$.

A.4.2 Application to Gaussian Processes

As seen in §6.2.1/p. 247, in the noise-free case, the Gaussian process prior specifies a joint normal prior over the training targets $\mathbf{f}$ and unknown test outputs $\mathbf{f}_*$,

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} K(\mathbf{X},\mathbf{X}) & K(\mathbf{X},\mathbf{X}_*) \\ K(\mathbf{X}_*,\mathbf{X}) & K(\mathbf{X}_*,\mathbf{X}_*) \end{bmatrix}\right),$$

where $\mathbf{X}$ is the matrix of training inputs, and $\mathbf{X}_*$ the matrix of test inputs. Applying the previous result (A.27) for conditioning a normal distribution, we can write the posterior distribution over the test outputs as

$$\mathbf{f}_* \sim \mathcal{N}\big(\boldsymbol{\mu}_{\mathbf{f}_*|\mathbf{f}}, \boldsymbol{\Sigma}_{\mathbf{f}_*|\mathbf{f}}\big),$$

with

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{f}_*|\mathbf{f}} &= K(\mathbf{X}_*,\mathbf{X})K(\mathbf{X},\mathbf{X})^{-1}\mathbf{f} \\ \boldsymbol{\Sigma}_{\mathbf{f}_*|\mathbf{f}} &= K(\mathbf{X}_*,\mathbf{X}_*) - K(\mathbf{X}_*,\mathbf{X})K(\mathbf{X},\mathbf{X})^{-1}K(\mathbf{X},\mathbf{X}_*), \end{aligned}$$

where we used the fact that the prior mean over both the training and test inputs is zero. The noisy case, where $\mathbf{y}|\mathbf{f} \sim \mathcal{N}(\mathbf{f}, \sigma_n^2 I_N)$, simply corresponds to adding a constant σ_n^2 to the diagonal of $K(\mathbf{X},\mathbf{X})$.

Glossary

ANOVA Analysis of Variance. (Page 191)

APT Arbitrage Pricing Theory. (Page 25)

ARIMA Autoregressive Integrated Moving Average. (Page 241)

BP Basis Point (one hundredth of one percent). (Page 129)

CAPM Capital Asset Pricing Model. (Page 25)

CARA Constant Absolute Risk Aversion. (Page 55)

CML Capital Market Line. (Page 12)

CRRA Constant Relative Risk Aversion. (Page 55)

DGP Data Generating Process. (Page 262)

DRAWDOWN Worst decline suffered by an investment from its peak value. (Page 4)

EFFICIENCY A portfolio $\mathbf{w}$ is said to be *efficient* if it is the lowest-variance portfolio for a given level of expected return. (Page 11)

FDA Functional Data Analysis. (Page 241)

FNNET Financial Neural Network. (Page 162)

GAUSSIAN PROCESS A collection of random variables, any finite number of which have a joint Gaussian distribution. (Page 245)

HARA Hyperbolic Absolute Risk Aversion. (Page 47)

IID Independent and Identically Distributed. (Page 40)

INTERACTION EFFECT Combined effect of two or more independent variables on a dependent variable when their joint effect is not additive. Used in analysis of variance tables as a correction to *main effects* to account for nonlinearities. (Page 192)

LEVERAGE (FINANCIAL) Ratio of asset value to the supporting equity.(Page 123)

LONG POSITION The buying of a security such as a stock, commodity or currency, with the expectation that the asset will rise in value. Opposite of *short position*. (Page 19)

MAIN EFFECT Effect of an independent variable (hyperparameter) on a dependent variable (e.g. classification accuracy), averaging over all levels of the other independent variables. Contrast with *interaction effect.* (Page 192)

MDP Markov Decision Process. (Page 134)

MLP Multi-Layer Perceptron. (Page 71)

MPT Modern Portfolio Theory. (Page 2)

NLL Negative Log-Likelihood. (Page 250)

OLS Ordinary Least Squares. (Page 69)

POMDP Partially Observed Markov Decision Process. (Page 85)

REA Recursive Enumeration Algorithm. (Page 136)

SHORT POSITION The sale of a borrowed security such as a stock, commodity or currency with the expectation that the asset will fall in value. Opposite of *long position.* (Page 19)

SIMPLE RETURN The *simple rate of return* of an asset during period t is given by $R_t = \frac{P_t}{P_{t-1}} - 1$ where P_t is the price of the asset at time t. (Page 6)

SVM Support Vector Machine. (Page 70)

VAR Vector Autoregressive. (Page 27)

References

Abu-Mostafa, Y. S. (1995). Hints. *Neural Computation 7*(4), 639–671.

Abu-Mostafa, Y. S., A. F. Atiya, M. Magdon-Ismail, and H. White (2001, July). Special Issue on Neural Networks in Financial Engineering. *IEEE Transactions on Neural Networks 12*(4), 653–656.

Alexander, G. J. and A. M. Baptista (2002). Economic Implications of Using a Mean-VaR Model for Portfolio Selection. *Journal of Economic Dynamics & Control 26*, 1159–1193.

Andrews, D. W. K. (1991, May). Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimation. *Econometrica 59*(3), 817–58.

Andrews, D. W. K. and J. C. Monahan (1992, July). An Improved Heteroskedasticity and Autocorrelation Consistent Covariance Matrix Estimator. *Econometrica 60*(4), 953–966.

Arrow, K. J. (1965). *Aspects of the Theory of Risk-Bearing*. Yrjö Jahnsson Lectures. Helsinki, Finland: Yrjö Jahnsson Foundation.

Artzner, P., F. Delbaen, J.-M. Eber, and D. Heath (1999). Coherent Measures of Risk. *Mathematical Finance 3*, 203–228.

Atkinson, C., S. R. Pliska, and P. Wilmott (1997, March). Portfolio Management with Transaction Costs. *Proceedings of the Royal Society: Mathematical, Physical and Engineering Sciences 453*(1958), 551–562.

Aït-Sahalia, Y. and M. W. Brandt (2001, August). Variable Selection for Portfolio Choice. *Journal of Finance 56*(4), 1297–1351.

Aït-Sahalia, Y. and M. W. Brandt (2007, November). Consumption and Portfolio Choice with Option-Implied State Prices. Working Paper, Princeton University.

Bagnell, J. A., S. Kakade, A. Y. Ng, and J. Schneider (2004). Policy Search by Dynamic Programming. In S. Thrun, L. Saul, and B. Schölkopf (Eds.), *Advances in Neural Information Processing Systems 16*, Cambridge, MA. MIT Press.

Banz, R. W. (1981). The Relationship Between Return and Market Value of Common Stocks. *Journal of Financial Economics 9*(1), 3–18.

Barberis, N. (2000). Investing for the Long Run When Returns Are Predictable. *Journal of Finance 55*(1), 225–264.

Barra (1998). *Risk Model Handbook, United States Equity: Version 3*. Berkeley, CA: MSCI Barra.

Basu, S. (1983). The Relationship Between Earnings Yield, Market Value, and Return for NYSE Common Stocks: Further Evidence. *Journal of Financial Economics 12*, 129–156.

Bauer, D. F. (1972, September). Constructing Confidence Sets Using Rank Statistics. *Journal of the American Statistical Association 67*(339), 687–690.

Beitelspacher, J., J. Fager, G. Henriques, and A. McGovern (2006, February). Policy Gradient vs. Value Function Approximation: A Reinforcement Learning Shootout. Technical report, School of Computer Science, University of Oklahoma, Norman, OK. Technical Report No. CS-TR-06-001.

Bellman, R. E. (1957). *Dynamic Programming*. Princeton, NJ: Princeton University Press.

Bellman, R. E. and S. E. Dreyfus (1959). Functional Approximation and Dynamic Programming. *Mathematical Tables and Other Aids to Computation 13*, 247–251.

Bellman, R. E., I. Glicksberg, and O. Gross (1955). On the Optimal Inventory Equation. *Management Science 2*, 83–104.

Bellman, R. E. and R. Kalaba (1960, December). On k th Best Policies. *Journal of the Society for Industrial and Applied Mathematics 8*(4), 582–588.

Ben-Tal, A. and A. Nemirovski (1999). Robust Solutions of Uncertain Linear Programs. *Operations Research Letters 25*, 1–13.

Bengio, Y. (1997). Using a Financial Training Criterion Rather than a Prediction Criterion. *International Journal of Neural Systems 8*(4), 433–443.

Bengio, Y., P. Lamblin, D. Popovici, and H. Larochelle (2007). Greedy Layer-Wise Training of Deep Networks. In B. Schölkopf, J. Platt, and T. Hoffman (Eds.), *Advances in Neural Information Processing Systems 19*, Cambridge, MA, pp. 153–160. MIT Press.

Bengio, Y., P. Simard, and P. Frasconi (1994). Learning Long-Term Dependencies with Gradient Descent is Difficult. *IEEE Transactions on Neural Networks 5*(2), 157–166. Special Issue on Recurrent Neural Networks.

Benveniste, A., M. Metivier, and P. Priouret (1990). *Adaptive Algorithms and Stochastic Approximations*. Springer.

Berkelaar, A. B., R. Kouwenberg, and T. Post (2004, November). Optimal Portfolio Choice under Loss Aversion. *Review of Economics and Statistics 86*(4), 973–987.

Bernoulli, D. (1738). Specimen theoriæ novæ de mensura sortis. In *Commentarii Academiæ Scientiarum Imperialis Petropolitannæ*. Translated from Latin into English by L. Sommer, "Exposition of a New Theory on the Measurement of Risk," Econometrica 22, 23–36.

Bertsekas, D. P. (2000). *Nonlinear Programming* (Second ed.). Belmont, MA: Athena Scientific.

Bertsekas, D. P. (2005). *Dynamic Programming and Optimal Control* (Third ed.), Volume I. Belmont, MA: Athena Scientific.

Bertsekas, D. P. (2007). *Dynamic Programming and Optimal Control* (Third ed.), Volume II. Belmont, MA: Athena Scientific.

Bertsekas, D. P. and J. N. Tsitsiklis (1996). *Neuro-Dynamic Programming*. Belmont, MA: Athena Scientific.

Bertsimas, D. and D. A. Pachamanova (2008, January). Robust Multiperiod Portfolio Management in the Presence of Transaction Costs. *Computers and Operations Research 35*(1), 3–17.

Best, M. J. and R. Grauer (1991). On the Sensitivity of Mean-Variance Efficient Portfolios to Changes in Asset Means: Some Analytical and Computational Results. *Review of Financial Studies 4*, 315–342.

Bevan, A. and K. Winkelmann (1998). Using the Black-Litterman Global Asset Allocation Model: Three Years of Practical Experience. Technical report, Fixed Income Research, Goldman Sachs.

Birge, J. R. and F. Louveaux (1997). *Introduction to Stochastic Programming*. New York, NY: Springer.

Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. London, UK: Oxford University Press.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York, NY: Springer.

Black, F. and R. Litterman (1992, September). Global Portfolio Optimization. *Financial Analysts Journal 48*(5), 28–43.

Blum, A. and A. Kalai (1998). Universal Portfolios With And Without Transaction Costs. *Machine Learning 30*(1), 23–30.

Bodie, Z., A. Kane, and A. J. Marcus (2004). *Investments* (Sixth ed.). Boston, MA: Irwin McGraw-Hill.

Bollerslev, T. (1986). Generalized Autoregressive Conditional Heteroskedasticity. *Journal of Econometrics 31*, 307–327.

Booch, G., R. A. Maksimchuk, M. W. Engel, B. J. Young, J. Conallen, and K. A. Houston (2007). *Object-Oriented Analysis and Design with Applications* (Third ed.). Reading, MA, USA: Addison-Wesley.

Boser, B. E., I. M. Guyon, and V. N. Vapnik (1992). A Training Algorithm for Optimal Margin Classifiers. In D. Haussler (Ed.), *Proceedings of*

the 5th Annual ACM Workshop on Computational Learning Theory, pp. 144–152. ACM Press.

Box, G. E. P., W. G. Hunter, and J. S. Hunter (2005). *Statistics for Experimenters: Design, Innovation, and Discovery* (Second ed.). John Wiley & Sons.

Boyd, S. and L. Vandenberghe (2004). *Convex Optimization*. Cambridge, UK: Cambridge University Press.

Brandt, M. W. (1999). Estimating Portfolio and Consumption Choice: A Conditional Euler Equations Approach. *Journal of Finance 54*(5), 1609–1645.

Brandt, M. W. (2004, August). Portfolio Choice Problems. Technical report, Duke University, Fuqua School of Business. Working Paper.

Brandt, M. W., A. Goyal, P. Santa-Clara, and J. R. Stroud (2005). A Simulation Approach to Dynamic Portfolio Choice with an Application to Learning About Return Predictability. *Review of Financial Studies 18*(3), 831–871.

Brandt, M. W. and P. Santa-Clara (2006, October). Dynamic Portfolio Selection by Augmenting the Asset Space. *Journal of Finance 61*(5), 2187–2217.

Brandt, M. W., P. Santa-Clara, and R. Valkanov (2007, September). Parametric Portfolio Policies: Exploiting Characteristics in the Cross Section of Equity Returns. Working Paper, Duke University.

Breiman, L. (1996). Bagging Predictors. *Machine Learning 24*(2), 123–140.

Brennan, M. J., E. S. Schwartz, and R. Lagnado (1997). Strategic Asset Allocation. *Journal of Economic Dynamics and Control 21*, 1377–1403.

Brown, K. C., W. V. Harlow, and L. T. Starks (1996, March). Of Tournaments and Temptations: An Analysis of Managerial Incentives in the Mutual Fund Industry. *Journal of Finance 51*(1), 85–110.

Brown, S. J. (1978). The Portfolio Choice Problem: Comparison of Certainty Equivalent and Optimal Bayes Portfolios. *Communications in Statistics: Simulation and Computation B7*, 321–334.

Butterworth, D. and P. Holmes (2002). Inter-market spread trading: evidence from UK index futures markets. *Applied Financial Economics 12*(11), 783–790.

Campbell, J. Y. (1991, March). A Variance Decomposition for Stock Returns. *Economic Journal 101*(405), 157–179.

Campbell, J. Y., A. W. Lo, and A. C. MacKinlay (1997). *The Econometrics of Financial Markets*. Princeton, NJ, USA: Princeton University Press.

Campbell, J. Y. and R. J. Shiller (1988). The Dividend Price Ratio and Expectations of Future Dividends And Discount Factors. *Review of Financial Studies 1*(3), 195–228.

Campbell, J. Y. and L. M. Viceira (1999, May). Consumption and Portfolio Decisions When Expected Returns Are Time Varying. *Quarterly Journal of Economics 114*(2), 433–495.

Campbell, J. Y. and L. M. Viceira (2002). *Strategic Asset Allocation: Portfolio Choice for Long-Term Investors*. New York, NY: Oxford University Press.

Carhart, M. M. (1997, March). On Persistence in Mutual Fund Performance. *Journal of Finance 52*(1), 57–82.

Chang, H. S., M. C. Fu, and J. H. S. I. Marcus (2007). *Simulation-based Algorithms for Markov Decision Processes*. Communications and Control Engineering Series. London, UK: Springer Verlag.

Chang, T.-J., N. Meade, J. E. Beasley, and Y. M. Sharaiha (2000). Heuristics for Cardinality Constrained Portfolio Optimization. *Computers and Operations Research 27*, 1271–1302.

Chapados, N. (2000). Critères d'optimisation d'algorithmes d'apprentissage en gestion de portefeuille. Master's thesis, Département d'informatique et de recherche opérationnelle, Université de Montréal, Montréal, Canada.

Chapados, N. and Y. Bengio (2001, July). Cost Functions and Model Combination for VaR-based Asset Allocation Using Neural Networks. *IEEE Transactions on Neural Networks 12*(4), 890–906.

Chapados, N. and Y. Bengio (2006). The K Best-Paths Approach to Approximate Dynamic Programming with Application to Portfolio Optimization. In L. Lamontagne and M. Marchand (Eds.), *Advances in Artificial Intelligence, 19th Conference of the Canadian Society for Computational Studies of Intelligence, Canadian AI 2006, Québec City, Québec, Canada, June 7–9, 2006, Proceedings*, Volume 4013 of *Lecture Notes in Computer Science*, pp. 491–502. Springer.

Chapados, N. and Y. Bengio (2007a, June). Forecasting Commodity Contract Spreads with Gaussian Processes. Presented at the 13th International Conference on Computing in Economics and Finance. Working paper available at `https://editorialexpress.com/cgi-bin/conference/download.cgi?db_name=sce2007&paper_id=98`.

Chapados, N. and Y. Bengio (2007b, February). Noisy K Best-Paths for Approximate Dynamic Programming with Application to Portfolio Optimization. *Journal of Computers 2*(1), 12–19.

Chapados, N. and Y. Bengio (2008). Augmented Functional Time Series Representation and Forecasting with Gaussian Processes. In J. C.

Platt, D. Koller, Y. Singer, and S. Roweis (Eds.), *Advances in Neural Information Processing Systems 20*, Cambridge, MA, pp. 265–272. MIT Press.

Chapados, N. and Y. Bengio (2009a). Forecasting and Trading Commodity Contract Spreads with Gaussian Processes. *International Journal of Forecasting*. Submitted for publication.

Chapados, N. and Y. Bengio (2009b). Training Graphs of Learning Modules for Sequential Data. *ACM Transactions on Knowledge Discovery from Data*. Submitted for publication.

Choey, M. and A. S. Weigend (1997). Nonlinear Trading Models Through Sharpe Ratio Maximization. In A. S. Weigend, Y. Abu-Mostafa, and A.-P. Refenes (Eds.), *Decision Technologies for Financial Engineering: Proceedings of the Fourth International Conference on Neural Networks in the Capital Markets (NNCM '96)*, pp. 3–22. World Scientific Publishing.

Chopra, V. K. and W. T. Ziemba (1993). The effect of errors in means, variances, and covariances on optimal portfolio choice. *Journal of Portfolio Management 19*(2), 6–11.

Chow, G. and M. Kritzman (2002, Spring). Value at Risk for Portfolios with Short Positions. *Journal of Portfolio Management ???*, 73–81.

Christoffersen, P. F. and F. X. Diebold (2006, August). Financial Asset Returns, Direction-of-Change Forecasting, and Volatility Dynamics. *Management Science 52*(8), 1273–1287.

Christoffersen, P. F., F. X. Diebold, R. S. Mariano, A. S. Tay, and Y. K. Tse (2007). Direction-of-Change Forecasts Based on Conditional Variance, Skewness and Kurtosis Dynamics: International Evidence. *Journal of Financial Forecasting 1*(2), 1–22.

Chu, W. and Z. Ghahramani (2005). Gaussian Processes for Ordinal Regression. *Journal of Machine Learning Research 6*, 1019–1041.

Claus, J. and J. Thomas (2001, October). Equity Premia as Low as Three Percent? Evidence from Analysts' Earnings Forecasts for Domestic and International Stock Markets. *Journal of Finance 56*(5), 1629–1666.

Clements, M. P. and D. F. Hendry (1998). *Forecasting Economic Time Series*. The Marshall Lectures on Economic Forecasting. Cambridge, UK: Cambridge University Press.

Clements, M. P. and D. F. Hendry (1999). *Forecasting Non-stationary Economic Time Series*. Zeuthen Lecture Book Series. Cambridge, MA: MIT Press.

Collins, A. (2006). *Beating the Financial Futures Market: Combining Small Biases into Powerful Money Making Strategies*. John Wiley & Sons.

Consigli, G. (2004). Estimation of Tail Risk and Portfolio Optimisation with Respect to Extreme Measures. In G. Szegö (Ed.), *Risk Measures for the 21st Century*, Chichester, pp. Chapter 18. John Wiley & Sons.

Cootner, P. H. (Ed.) (1964). *The Random Character of Stock Market Prices*. Cambridge, MA: MIT Press.

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein (2001). *Introduction to Algorithms* (Second ed.). Cambridge, MA: MIT Press.

Cover, T. M. (1991). Universal Portfolios. *Mathematical Finance 1*(1), 1–29.

Cover, T. M. and E. Ordentlich (1996). Universal Portfolios With Side Information. *IEEE Transactions on Information Theory 42*(2), 348–363.

Cover, T. M. and J. A. Thomas (2006). *Elements of Information Theory* (Second ed.). New York, NY: John Wiley & Sons.

Cox, J. C. and C.-F. Huang (1989). Optimum Consumption And Portfolio Policies When Asset Prices Follow A Diffusion Process. *Journal of Economic Theory 49*, 33–83.

Cox, J. C. and C.-F. Huang (1991). A Variational Problem Arising in Financial Economics. *Journal of Mathematical Economics 20*, 465–487.

Cressie, N. A. C. (1993). *Statistics for Spatial Data*. Wiley.

Crichfield, T., T. Dyckman, and J. Lakonishok (1978, July). An Evaluation of Security Analysts' Forecasts. *The Accounting Review 53*(3), 651–668.

Croarkin, C. and P. Tobias (Eds.) (2006). *NIST/Sematech e-Handbook of Statistical Methods*. U.S. Commerce Department. Available at `http://www.itl.nist.gov/div898/handbook/index.htm`.

Cvitanić, J. (2001). Theory of Portfolio Optimization in Markets with Frictions. In E. Jouini, J. Cvitanić, and M. Musiela (Eds.), *Handbooks in Mathematical Finance: Option Pricing, Interest Rates and Risk Management*, Cambridge, UK, pp. 577–631. Cambridge University Press.

Cvitanić, J., L. Goukasian, and F. Zapatero (2003, April). Monte Carlo Computation Of Optimal Portfolios In Complete Markets. *Journal of Economic Dynamics and Control 27*(6), 971–986.

Cvitanić, J. and I. Karatzas (1996). Hedging and Portfolio Optimization Under Transaction Costs: A Martingale Approach. *Mathematical Finance 6*(2), 133–165.

Cvitanić, J., A. Lazrak, and T. Wang (2008, May). Implications Of The Sharpe Ratio As A Performance Measure In Multi-Period Settings. *Journal of Economic Dynamics and Control 32*(5), 1622–1649.

Dammon, R. M., C. S. Spatt, and H. H. Zhang (2001). Optimal Consumption and Investment with Capital Gains Taxes. *Review of Financial Studies 14*(3), 583–616.

Dantzig, G. B. (1955, April–July). Linear Programming Under Uncertainty. *Management Science 1*(3–4), 197–206.

Dantzig, G. B. and G. Infanger (1993). Multi-Stage Stochastic Linear Programs For Portfolio Optimization. *Annals of Operations Research 45*(1-4), 59–76.

Davis, M. H. A. and A. R. Norman (1990, November). Portfolio Selection with Transaction Costs. *Mathematics of Operations Research 15*(4), 676–713.

Dawid, A. P. (1984). Present position and potential developments: some personal views. Statistical theory. The prequential approach (with Discussion). *Journal of the Royal Statistical Society A 147*, 278–292.

Dawid, A. P. (1992). Prequential Analysis, Stochastic Complexity and Bayesian Inference (with Discussion). In J. M. Bernardo, J. O. Berger, A. P. Dawid, and A. F. M. Smith (Eds.), *Bayesian Statistics 4*, pp. 109–125. Oxford University Press.

De Bondt, W. F. M. and R. Thaler (1985, July). Does the Stock Market Overreact? *Journal of Finance 40*(3), 793–805.

DeMiguel, V., L. Garlappi, and R. Uppal (2009). Optimal Versus Naive Diversification: How Inefficient is the 1/N Portfolio Strategy? *Rev. Financ. Stud. 22*(5), 1915–1953.

DeMiguel, V. and R. Uppal (2005, February). Portfolio Investment with the Exact Tax Basis via Nonlinear Programming. *Management Science 51*(2), 277–290.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum Likelihood From Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, B 39*(1), 1–38.

Dempster, M. A. H., M. Germano, E. A. Medova, M. I. Rietbergen, F. Sandrini, and M. Scrowston (2007). Designing Minimum Guaranteed Return Funds. *Quantitative Finance 7*(2), 245–256.

Dempster, M. A. H., M. Germano, E. A. Medova, and M. Villaverde (2003). Global Asset Liability Management. *British Actuarial Journal 9*(1), 137–216.

Dempster, M. A. H., G. Pflug, and G. Mitra (Eds.) (2008). *Quantitative Fund Management*. Financial Mathematics Series. Chapman & Hall / CRC.

Detemple, J. B., R. Garcia, and M. Rindisbacher (2003, February). A Monte Carlo Method for Optimal Portfolios. *Journal of Finance 58*(1), 401–446.

Diebold, F. X. and R. S. Mariano (1995, July). Comparing Predictive Accuracy. *Journal of Business & Economic Statistics 13*(3), 253–263.

Dreyfus, S. E. (1969). An Appraisal Of Some Shortest Path Algorithms. *Operations Research 17*, 395–412.

Duffie, D. (2001). *Dynamic Asset Pricing Theory* (Third ed.). Princeton, NJ: Princeton University Press.

Dunis, C. L., J. Laws, and B. Evans (2006a). Modelling And Trading The Soybean-Oil Crush Spread With Recurrent And Higher Order Networks: A Comparative Analysis. *Neural Network World 16*, 193–213.

Dunis, C. L., J. Laws, and B. Evans (2006b, January). Trading Futures Spread Portfolios: Applications of Higher Order and Recurrent Networks. Technical report, Centre for International Banking, Economics and Finance; Liverpool John Moores University. `www.cibef.com`.

Dunis, C. L., J. Laws, and B. Evans (2006c). Trading Futures Spreads: an Application of Correlation and Threshold Filters. *Applied Financial Economics 16*(12), 903–914.

Dunis, C. L., J. Laws, and P. Naïm (Eds.) (2003). *Applied Quantitative Methods for Trading and Investment*. Chichester, UK: John Wiley & Sons.

Dunis, C. L. and J. Miao (2006). Volatility Filters for Asset Management: An Application to Managed Futures. *Journal of Asset Management 7*, 179–189.

Dutt, H. R., J. Fenton, J. D. Smith, and G. H. Wang (1997). Crop Year Influences and Variability of the Agricultural Futures Spreads. *The Journal of Futures Markets 17*(3), 341–367.

Eastham, J. F. and K. J. Hastings (1988, November). Optimal Impulse Control of Portfolios. *Mathematics of Operations Research 13*(4), 588–605.

Edwards, E. O. and P. W. Bell (1961). *Theory and Measurement of Business Income*. Berkeley, CA: University of California Press.

Elton, E. J. and M. J. Gruber (1978). Taxes and Portfolio Composition. *Journal of Financial Economics 6*, 399–410.

Engle, R. (1982). Autoregressive Conditional Heteroscedasticity, with Estimates of the Variance of United Kingdom Inflation. *Econometrica 50*, 987–1007.

Eppstein, D. (1998). Finding the k Shortest Paths. *SIAM Journal on Computing 28*(2), 652–673.

Epstein, L. G. and S. E. Zin (1989, July). Substitution, Risk Aversion, and the Temporal Behavior of Consumption and Asset Returns: A Theoretical Framework. *Econometrica 57*(4), 937–969.

Erb, C. B. and C. R. Harvey (2006, March/April). The Strategic and Tactical Value of Commodity Futures. *Financial Analysts Journal 62*(2), 69–97.

Estrada, J. (2007, November). Mean-Semivariance Optimization: A Heuristic Approach. Technical report, IESE Business School, Spain.

Fabozzi, F. J., S. M. Focardi, and P. N. Kolm (2006). *Financial Modeling of the Equity Market: From CAPM to Cointegration*. The Frank J. Fabozzi Series. Hoboken, NJ: John Wiley & Sons.

Fabozzi, F. J., P. N. Kolm, D. A. Pachamanova, and S. M. Focardi (2007). *Robust Portfolio Optimization and Management*. The Frank J. Fabozzi Series. Hoboken, NJ: John Wiley & Sons.

Fama, E. F. (1970, May). Efficient Capital Markets: A Review of Theory and Empirical Work. *Journal of Finance 25*(2), 383–417. Papers and Proceedings of the Twenty-Eighth Annual Meeting of the American Finance Association New York, N.Y. December, 28–30, 1969.

Fama, E. F. and K. R. French (1988). Dividend Yields And Expected Stock Returns. *Journal of Financial Economics 22*(1), 3–25.

Fama, E. F. and K. R. French (1992, June). The Cross-Section of Expected Stock Returns. *Journal of Finance 47*(2), 427–465.

Fama, E. F. and K. R. French (1993). Common Risk Factors in the Returns on Stocks and Bonds. *Journal of Financial Economics 33*, 3–56.

Fama, E. F. and K. R. French (1995). Size and Book-to-Market Factors in Earnings and Returns. *Journal of Finance 50*(1), 131–155.

Fama, E. F. and K. R. French (1996, March). Multifactor Explanations of Asset Pricing Anomalies. *Journal of Finance 51*(1), 55–84.

Fama, E. F. and K. R. French (2002). The Equity Premium. *The Journal of Finance 57*(2), 637–659.

Farinelli, S., D. Rossello, and L. Tobiletti (2006). Computational Asset Allocation Using One-Sided and Two-Sided Variability Measures. In *Computational Science – ICCS 2006*, Lecture Notes in Computer Science 3994, Berlin / Heidelberg, pp. 324–331. Springer.

Fisher, I. (1906). *The Nature of Capital and Income*. London: Macmillan.

Fisher, K. L. (2006). *The Only Three Questions That Count: Investing by Knowing What Others Don't*. John Wiley & Sons.

Fox, E. B., E. B. Sudderth, M. I. Jordan, and A. S. Willsky (2009). Nonparametric Bayesian Learning of Switching Linear Dynamical Systems. In *Advances in Neural Information Processing Systems 21: Proceedings of the 2008 Conference*, Cambridge, MA. MIT Press. To Appear.

Freund, Y. and R. E. Schapire (1996). Experiments with a new Boosting algorithm. In *Machine Learning: Proceedings of Thirteenth International Conference*, USA, pp. 148–156. ACM.

Friesen, G. and P. A. Weller (2006, November). Quantifying Cognitive Biases in Analyst Earnings Forecasts. *Journal of Financial Markets 9*(4), 333–365.

Frost, P. A. and J. E. Savarino (1986). An Empirical Bayes Approach to Efficient Portfolio Selection. *Journal of Financial and Quantitative Analysis 21*, 293–305.

Frost, P. A. and J. E. Savarino (1988). For Better Performance: Constrain Portfolio Weights. *Journal of Portfolio Management 15*, 29–34.

Fung, W. and D. A. Hsieh (2001). The Risk in Hedge Fund Strategies: Theory and Evidence from Trend Followers. *Review of Financial Studies 14*(2), 313–341.

Gallmeyer, M. F., R. Kaniel, and S. Tompaidis (2006, May). Tax Management Strategies With Multiple Risky Assets. *Journal of Financial Economics 80*(2), 243–291.

Geman, S., E. Bienenstock, and R. Doursat (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation 4*(1), 1–58.

Ghahramani, Z. and G. E. Hinton (1996). Parameter Estimation for Linear Dynamical Systems. Technical Report CRG-TR-96-2, Department of Computer Science, University of Toronto.

Ghosn, J. and Y. Bengio (1997). Multi-Task Learning for Stock Selection. In M. I. Jordan, M. C. Mozer, and T. Petsche (Eds.), *Advances in Neural Information Processing Systems 9*, pp. 946–952. MIT Press, Cambridge, MA.

Gingras, F., Y. Bengio, and C. Nadeau (2002). On Out-of-Sample Statistics for Time-Series. Technical Report 2002s-51, CIRANO. Available at `http://www.cirano.qc.ca/pdf/publication/2002s-51.pdf`.

Girard, A., C. E. Rasmussen, J. Q. Candela, and R. Murray-Smith (2003). Gaussian Process Priors with Uncertain Inputs – Application to Multiple-Step Ahead Time Series Forecasting. In S. T. S. Becker and K. Obermayer (Eds.), *Advances in Neural Information Processing Systems 15*, pp. 529–536. MIT Press.

Girma, P. B. and A. S. Paulson (1998). Seasonality in Petroleum Futures Spreads. *The Journal of Futures Markets 18*(5), 581–598.

Givoly, D. and J. Lakonishok (1984, September–October). The Quality of Analysts' Forecasts of Earnings. *Financial Analysts Journal 40*(5), 40–47.

Glynn, P. W. (1986). Stochastic Approximation for Monte Carlo Optimization. In *Proceedings of the 1986 Winter Simulation Conference*, pp. 285–289.

Goldberg, P. W., C. K. I. Williams, and C. M. Bishop (1998). Regression with Input-Dependent Noise: A Gaussian Process Treatment. In

M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems 10: Proceedings of the 1997 Conference*, Cambridge, MA, pp. 493–499. MIT Press.

Goldfarb, D. and G. Iyengar (2003). Robust Portfolio Selection Problems. *Mathematics of Operations Research 28*, 1–38.

Gordon, M. (1962). *The Investment, Financing, and Valuation of the Corporation*. Homewood, IL: Irwin Publishing.

Graham, B. and D. L. Dodd (1934). *Security Analysis: Principles and Techniques*. Columbus, OH: McGraw-Hill.

Greene, W. H. (2007). *Econometric Analysis* (Sixth ed.). Englewood Cliffs, NJ: Prentice Hall.

Grinold, R. C. and R. N. Kahn (2000). *Active Portfolio Management*. McGraw Hill.

Grothendieck, G. (2005). *dyn: Time Series Regression*. R package version 0.2-6.

Halldórsson, B. V. and R. H. Tütüncü (2003). An Interior-Point Method for a Class of Saddle-Point Problems. *Journal of Optimization Theory and Applications 116*(3), 559–590.

Hamilton, J. D. (1994). *Time Series Analysis*. Princeton, NJ: Princeton University Press.

Hansen, P. R. and A. Lunde (2005). A forecast comparison of volatility models: does anything beat a GARCH(1,1)? *Journal of Applied Econometrics 20*(7), 873–889.

Harrison, M. and D. Kreps (1979). Martingales and Arbitrage in Multiperiod Securities Markets. *Journal of Economic Theory 20*, 381–408.

Hart, P. E., N. J. Nilsson, and B. Raphael (1968, July). A Formal Basis for the Heuristic Determination of Minimum Cost Paths. *IEEE Transactions on Systems Science and Cybernetics 4*(2), 100–107.

Hartmann, A. K. and M. Weigt (2005). *Phase Transitions in Combinatorial Optimization Problems*. Weinheim, Germany: Wiley-VCH Verlag.

Hastie, T., R. Tibshirani, and J. Friedman (2001). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Berlin, New York: Springer.

Hautsch, N. and D. Hess (2002, January). The Processing of Non-Anticipated Information in Financial Markets: Analyzing the Impact of Surprises in the Employment Report. *European Finance Review 6*(2), 133–161.

Helmbold, D. P., R. E. Schapire, Y. Singer, and M. K. Warmuth (1998). On-line Portfolio Selection Using Multiplicative Updates. *Mathematical Finance 8*(4), 325–347.

Hens, T. and P. Wöhrmann (2007). Strategic Asset Allocation And Market Timing: A Reinforcement Learning Approach. *Computational Economics 29*, 369–381.

Herbster, M. and M. K. Warmuth (1998). Tracking the Best Expert. *Machine Learning 32*(2), 151–178.

Hess, D. (2001). Surprises in U.S. Macroeconomic Releases: Determinants of Their Relative Impact on T-Bond Futures. Technical report, Center of Finance and Econometrics, University of Konstanz. Discussion Paper No. 2001/01. Available at SSRN: `http://ssrn.com/abstract=267492`.

Hinton, G. E., S. Osindero, and Y.-W. Teh (2006). A Fast Learning Algorithm for Deep Belief Nets. *Neural Computation 18*(7), 1527–1554.

Hoerl, A. E. and R. W. Kennard (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics 12*(3), 55–67.

Hornik, K., M. B. Stinchcombe, and H. White (1989). Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks 2*, 359–366.

Hsu, J. (1996). *Multiple Comparisons: Theory and Methods*. Boca Raton, FL: Chapman & Hall/CRC.

Hull, J. C. (2005). *Options, Futures and Other Derivatives* (Sixth ed.). Englewood Cliffs, NJ: Prentice Hall.

Härdle, W. (1990). *Applied Nonparametric Regression*. New York, NY: Cambridge University Press.

Itô, K. (1951). On Stochastic Differential Equations. *Memoirs of the American Mathematical Sociecy 4*, 1–51.

Jaakkola, T., S. P. Singh, and M. I. Jordan (1995). Reinforcement Learning Algorithm For Partially Observable Markov Decision Problems. In G. Tesauro, D. S. Touretzky, and T. K. Leen (Eds.), *Advances in Neural Information Processing Systems 7*, Cambridge, MA, pp. 345–352. MIT Press.

Jagannathan, R. and T. Ma (2003). Risk Reduction in Large Portfolios: Why Imposing the Wrong Constraints Helps. *Journal of Finance 58*, 1651–1683.

James, W. and C. Stein (1961). Estimation With Quadratic Loss. In *Proceedings of the Fourth Berkeley Symposium on Mathematics and Statistics*, pp. 361–379.

Jarque, C. M. and A. K. Bera (1980). Efficient Tests for Normality, Homoscedasticity and Serial Independence Of Regression Residuals. *Economics Letters 6*(3), 255–259.

Jegadeesh, N. and S. Titman (1993, March). Returns to Buying Winners and Selling Losers: Implications for Stock Market Efficiency. *Journal of Finance 48*(1), 65–91.

Jiménez, V. M. and A. Marzal (1999). Computing the K Shortest Paths: A New Algorithm and an Experimental Comparison. In *WAE '99: Proceedings of the 3rd International Workshop on Algorithm Engineering*, Volume 1668 of *Lecture Notes in Computer Science*, London, UK, pp. 15–29. Springer-Verlag.

Jiménez, V. M. and A. Marzal (2003). A Lazy Version of Eppstein's K Shortest Paths Algorithm. In *Experimental and Efficient Algorithms: Second International Workshop, WEA 2003, Ascona, Switzerland, May 26-28, 2003, Proceedings*, Volume 2647 of *Lecture Notes in Computer Science*, pp. 179–190. Springer.

Jin, H., H. M. Markowitz, and X. Y. Zhou (2006, January). A Note on Semivariance. *Mathematical Finance 16*(1), 53–61.

Jobson, J. D. and B. M. Korkie (1980). Estimation of Markowitz Efficient Portfolios. *Journal of the American Statistical Association 75*, 544–554.

Jobson, J. D. and B. M. Korkie (1981, September). Performance Hypothesis Testing with the Sharpe and Treynor Measures. *Journal of Finance 36*(4), 889–908.

Jobson, J. D. and V. Ratti (1979). Improved estimation for Markowitz portfolios using James-Stein type estimators. In *Proceedings of the American Statistical Association, Business and Economic Statistics Section*, pp. 279–284.

Jorion, P. (1986). Bayes-Stein Estimation for Portfolio Analysis. *Journal of Financial and Quantitative Analysis 21*, 279–292.

Jorion, P. (2003). Portfolio Optimization with Tracking-Error Constraints. *Financial Analysts Journal 59*, 70–82.

Kahneman, D. and A. Tversky (1979, March). Prospect Theory: An Analysis of Decision under Risk. *Econometrica 47*(2), 263–292.

Kailath, T., A. H. Sayed, and B. Hassibi (2000). *Linear Estimation*. Englewood Cliffs, NJ: Prentice Hall.

Kaizoji, T., S. Bornholdt, and Y. Fujiwara (2002, December). Dynamics of Price and Trading Volume In A Spin Model Of Stock Markets With Heterogeneous Agents. *Physica A 316*(1–4), 441–452.

Kalai, A. and S. Vempala (2002). Efficient Algorithms for Universal Portfolios. *Journal of Machine Learning Research 3*, 423–440.

Kallberg, J. G. and W. T. Ziemba (1983, November). Comparison of Alternative Utility Functions in Portfolio Selection Problems. *Management Science 29*(11), 1257–1276.

Kalman, R. E. (1960). A New Approach to Linear Filtering and Prediction Problems. *Transactions of the American Society of Mechanical Engineering, Series D, Journal of Basic Engineering 82*, 34–45.

Kandel, S. and R. F. Stambaugh (1996, June). On the Predictability of Stock Returns: An Asset-Allocation Perspective. *Journal of Finance 51*(2), 385–424.

Karatzas, I., J. P. Lehoczky, and S. E. Shreve (1987). Optimal portfolio and consumption decisions for a "small investor" on a finite horizon. *SIAM Journal of Control and Optimization 25*(6), 1557–1586.

Kaufman, P. J. (1998). *Trading Systems and Methods* (Third ed.). John Wiley & Sons.

Kearns, M. J., Y. Mansour, and A. Y. Ng (2000). Approximate Planning in Large POMDPs Via Reusable Trajectories. In *Advances in Neural Information Processing Systems 12*, Cambridge, MA, pp. 1001–1007. MIT Press.

Kellerer, H., R. Mansini, and M. G. Speranza (2000). Selecting Portfolios with Fixed Costs and Minimum Transaction Lots. *Annals of Operations Research 99*, 287–304.

Kim, M. K. and R. M. Leuthold (2000, April). The Distributional Behavior of Futures Price Spreads. *Journal of Agricultural and Applied Economics 32*(1), 73–87.

Kim, T. S. and E. Omberg (1996). Dynamic Nonmyopic Portfolio Behavior. *The Review of Financial Studies 9*(1), 141–161.

Kirkpatrick, S. and B. Selman (1994, May). Critical Behavior in the Satisfiability of Random Boolean Expressions. *Science 264*(5163), 1297–1301.

Kissell, R. and M. Glantz (2003). *Optimal Trading Strategies: Quantitative Approaches for Managing Market Impact and Trading Risk*. New York, NY: AMACOM/American Management Association.

Kivinen, J. and M. K. Warmuth (1997). Exponentiated Gradient versus Gradient Descent for Linear Predictors. *Information and Computation 132*(1), 1–63.

Klein, R. and V. Bawa (1976). The Effect of Estimation Risk on Optimal Portfolio Choice. *Journal of Financial Economics 3*, 215–231.

Konda, V. R. and J. N. Tsitsiklis (2003). On Actor-Critic Algorithms. *SIAM Journal on Control and Optimization 42*(4), 1143–1166.

Koo, H. K. (1998). Consumption and Portfolio Selection with Labor Income: A Continuous Time Approach. *Mathematical Finance 8*(1), 49–65.

Korn, R. and S. Trautmann (1995). Continuous-Time Portfolio Optimization Under Terminal Wealth Constraints. *ZOR – Mathematical Methods of Operations Research 42*(1), 69–92.

Krokhmal, P., J. Palmquist, and S. Uryasev (2002). Portfolio Optimization with Conditional Value-at-Risk Objective and Constraints. *The Journal of Risk 4*(2), 11–27.

Lagoudakis, M. G. and R. Parr (2003). Reinforcement Learning as Classification: Leveraging Modern Classifiers. In *in Proceedings of the Twentieth International Conference on Machine Learning*, pp. 424–431.

Lahiri, S. N. (2003). *Resampling Methods for Dependent Data*. New York, NY: Springer-Verlag.

Lakonishok, J., A. Shleifer, and R. W. Vishny (1994, December). Contrarian Investment, Extrapolation, and Risk. *Journal of Finance 49*(5), 1541–1578.

Laloux, L., P. Cizeau, J.-P. Bouchaud, and M. Potters (1999). Noise Dressing of Financial Correlation Matrices. *Physics Review Letter 83*, 1467–1470.

Langford, J. and B. Zadrozny (2005, August). Relating Reinforcement Learning Performance to Classification Performance. In *22nd International Conference on Machine Learning (ICML 2005)*, Bonn, Germany.

Law, A. M. and W. D. Kelton (2000). *Simulation Modeling and Analysis* (Third ed.). McGraw Hill.

Ledoit, O. and M. Wolf (2004). Honey, I Shrunk the Sample Covariance Matrix. *Journal of Portfolio Management 30*(4), 110–119.

Ledoit, O. and M. Wolf (2008). Robust Performance Hypothesis Testing with the Sharpe Ratio. *Journal of Empirical Finance 15*, 850–859.

Leippold, M., F. Trojani, and P. Vanini (2004, March). A Geometric Approach To Multiperiod Mean Variance Optimization Of Assets And Liabilities. *Journal of Economic Dynamics and Control 28*(6), 1079–1113.

Leland, H. E. (2000). Optimal Portfolio Implementation With Transaction Costs and Capital Gains Taxes. Working Paper, University of California, Berkeley.

Levy, H. and H. M. Markowitz (1979, June). Approximating Expected Utility by a Function of Mean and Variance. *American Economic Review 69*(3), 308–317.

Li, D. and W.-L. Ng (2000, July). Optimal Dynamic Portfolio Selection: Multiperiod Mean-Variance Formulation. *Mathematical Finance 10*(3), 387–406.

Li, L. and H.-T. Lin (2007). Ordinal Regression by Extended Binary Classification. In B. Schölkopf, J. Platt, and T. Hoffman (Eds.), *Advances in Neural Information Processing Systems 19*, pp. 865–872. Cambridge, MA: MIT Press.

Li, P., C. Burges, and Q. Wu (2008). McRank: Learning to Rank Using Multiple Classification and Gradient Boosting. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis (Eds.), *Advances in Neural Information Processing Systems 20*, pp. 897–904. Cambridge, MA: MIT Press.

Lintner, J. (1965, February). The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets. *The Review of Economics and Statistics 47*(1), 13–37.

Litterman, R. M. (Ed.) (2003). *Modern Investment Management*. Wiley Finance.

Liu, H. (2004, February). Optimal Consumption and Investment with Transaction Costs and Multiple Risky Assets. *Journal of Finance 59*(1), 289–338.

Liu, S.-M. and C.-H. Chou (2003). Parities and Spread Trading in Gold and Silver Markets: A Fractional Cointegration Analysis. *Applied Financial Economics 13*(12), 879–891.

Ljung, G. M. and G. E. P. Box (1978, August). On a Measure of Lack of Fit in Time Series Models. *Biometrika 65*(2), 297–303.

Lo, A. W. and A. C. MacKinlay (1990). Data-snooping biases in tests of financial asset pricing models. *Review of Financial Studies 3*(3), 431–467.

Lo, A. W. and A. C. MacKinlay (1999). *A Non-Random Walk Down Wall Street*. Princeton, NJ: Princeton University Press.

Longstaff, F. A. and E. S. Schwartz (2001). Valuing American Options by Simulation: A Simple Least-Squares Approach. *Review of Financial Studies 14*(1), 113–147.

Luenberger, D. G. and Y. Ye (2007). *Linear and Nonlinear Programming* (Third ed.). International Series in Operations Research & Management Science. Springer.

MacKay, D. J. C. (1994). Bayesian Methods for Backprop Networks. In E. Domany, J. L. van Hemmen, and K. Schulten (Eds.), *Models of Neural Networks, III*, pp. 211–254. Springer.

MacKay, D. J. C. (1999). Comparison of Approximate Methods for Handling Hyperparameters. *Neural Computation 11*(5), 1035–1068.

MacKay, D. J. C. (2003). *Information Theory, Inference, and Learning Algorithms*. Cambridge, U.K.: Cambridge University Press.

Magill, M. J. P. and G. M. Constantinides (1976, October). Portfolio Selection With Transactions Costs. *Journal Of Economic Theory 13*(2), 245–263.

Malevergne, Y. and D. Sornette (2005a). *Extreme Financial Risks: From Dependence to Risk Management*. Berlin / Heidelberg: Springer.

Malevergne, Y. and D. Sornette (2005b). High-Order Moments and Cumulants of Multivariate Weibull Asset Returns Distributions: Analytical Theory and Empirical Tests II. *Finance Letters, Special Issue: Modeling of the Equity Market 3*(1), 54–63.

Marbach, P. and J. N. Tsitsiklis (2001, February). Simulation-Based Optimization of Markov Reward Processes. *IEEE Transactions on Automatic Control 46*(2), 191–209.

Marbach, P. and J. N. Tsitsiklis (2003). Approximate Gradient Methods in Policy-Space Optimization of Markov Reward Processes. *Discrete Event Dynamic Systems: Theory and Applications 13*, 111–148.

Markowitz, H. M. (1952). Portfolio Selection. *Journal of Finance 7*, 77–91.

Markowitz, H. M. (1959). *Portfolio Selection: Efficient Diversification of Investment*. New York, London, Sydney: John Wiley & Sons.

Markowitz, H. M. (1999). The Early History of Portfolio Theory: 1600–1960. *Financial Analysts Journal 55*, 5–16.

Markowitz, H. M. and N. Usmen (2003). Resampled Frontiers versus Diffuse Bayes: An Experiment. *Journal of Investment Management 1*, 9–25.

Matheron, G. (1973). The Intrinsic Random Functions and Their Applications. *Advances in Applied Probability 5*, 439–468.

Mathieson, M. (1995). Ordinal Models for Risk Assessment. In *Proceedings of the U.K. Institute for Quantitative Investment Research, Autumn Seminar*, Hertfordshire, U.K.

Mathieson, M. (1997). Ordered Classes and Incomplete Examples in Classification. In M. C. Mozer, M. I. Jordan, and T. Petsche (Eds.), *Advances in Neural Information Processing Systems 9*, Cambridge, MA, pp. 550–556. MIT Press.

McCullagh, P. (1980). Regression Models for Ordinal Data (with discussion). *Journal of the Royal Statistical Society B 42*(2), 109–142.

McCullagh, P. and J. A. Nelder (1989). *Generalized Linear Models* (Second ed.). London: Chapman & Hall.

McCulloch, C. E., S. R. Searle, and J. M. Neuhaus (2008). *Generalized, Linear, and Mixed Models* (Second ed.). Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons.

McNelis, P. D. (2005). *Neural Networks in Finance: Gaining Predictive Edge in the Market*. Burlington, MA: Academic Press.

Memmel, C. (2003). Performance Hypothesis Testing with the Sharpe Ratio. *Finance Letters 1*, 21–23.

Merton, R. C. (1969). Lifetime Portfolio Selection Under Uncertainty: the Continuous-Time Case. *Review of Economics and Statistics 51*(3), 247–257. Reprinted in Merton (1990, Chapter 4).

Merton, R. C. (1971). Optimum Consumption and Portfolio Rules in a Continuous-Time Model. *Journal of Economic Theory 3*, 373–413. Reprinted in Merton (1990, Chapter 5).

Merton, R. C. (1972, September). An Analytic Derivation of the Efficient Portfolio Frontier. *The Journal of Financial and Quantitative Analysis 7*(4), 1851–1872.

Merton, R. C. (1973, September). An Intertemporal Capital Asset Pricing Model. *Econometrica 41*(5), 867–887. Reprinted in Merton (1990, Chapter 15).

Merton, R. C. (1990). *Continuous-Time Finance*. Cambridge, MA: Blackwell Publishers.

Merton, R. C. and P. A. Samuelson (1974, May). Fallacy of the Log-Normal Approximation to Optimal Portfolio Decision Making Over Many Periods. *Journal of Financial Economics 1*, 67–94.

Michaud, R. O. (1989). The Markowitz Optimization Enigma: Is Optimized Optimal? *Financial Analysts Journal 45*, 31–42.

Michaud, R. O. (1998). *Efficient Asset Management: A Practical Guide to Stock Portfolio Optimization and Asset Allocation*. Oxford, UK: Oxford University Press.

Miffre, J. and G. Rallis (2007, Forthcoming in Journal of Banking and Finance). Momentum Strategies in Commodity Futures Markets. Technical report, EDHEC Risk and Asset Management Research Centre, Nice, France.

Mitchell, D., B. Selman, and H. Levesque (1992, July). Hard and Easy Distributions of SAT Problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San Jose, CA, pp. 459–465.

Mittnik, S., S. T. Rachev, and E. S. Schwartz (2003). Value At Risk and Asset Allocation with Stable Return Distributions. *Allgemeines Statistisches Archiv 86*, 53–67.

Montier, J. (2002). *Behavioural Finance: Insights into Irrational Minds and Market*. Chichester, UK: John Wiley & Sons.

Moody, J. and M. Saffell (2001a, July). Learning to Trade via Direct Reinforcement. *IEEE Transactions on Neural Networks 12*(4), 875–889.

Moody, J. and M. Saffell (2001b). Learning to Trade via Direct Reinforcement. *IEEE Transactions on Neural Networks 12*(4), 875–889.

Moody, J., L. Wu, Y. Liao, and M. Saffell (1998). Performance Functions and Reinforcement Learning for Trading Systems and Portfolios. *Journal of Forecasting 17*, 441–470.

Morton, A. J. and S. R. Pliska (1995). Optimal Portfolio Management With Fixed Transaction Costs. *Mathematical Finance 5*(4), 337–356.

Mossin, J. (1966, October). Equilibrium in a Capital Asset Market. *Econometrica 34*(4), 768–783.

Mossin, J. (1968). Optimal Multiperiod Portfolio Policies. *Journal of Business 41*(2), 215–229.

Muthuraman, K. and H. Zha (2008). Simulation-Based Portfolio Optimization For Large Portfolios With Transaction Costs. *Mathematical Finance 18*(1), 115–134.

Mézard, M., G. Parisi, and M. A. Virasoro (1987). *Spin Glass Theory and Beyond*. Singapore: World Scientific.

Nawrocki, D. (1999). A Brief History of Downside Risk Measures. *Journal of Investing 8*, 9–25.

Neal, R. M. (1996). *Bayesian Learning for Neural Networks*. Lecture Notes in Statistics 118. Springer.

Neal, R. M. and G. E. Hinton (1998). A View of the EM Algorithm that Justifies Incremental, Sparse and Other Variants. In M. I. Jordan (Ed.), *Learning in Graphical Models*, Cambridge, MA, pp. 355–368. MIT Press.

Neuneier, R. (1996). Optimal Asset Allocation using Adaptive Dynamic Programming. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems 8*, pp. 952–958. MIT Press.

Neuneier, R. (1998). Enhancing Q-Learning for Optimal Asset Allocation. In M. I. Jordan, M. J. Kearns, and S. A. Solla (Eds.), *Advances in Neural Information Processing Systems 10*, pp. 936–942. The MIT Press.

Neuneier, R. and O. Mihatsch (1999). Risk Sensitive Reinforcement Learning. In M. J. Kearns, S. A. Solla, and D. A. Cohn (Eds.), *Advances in Neural Information Processing Systems 11*, pp. 1031–1037. The MIT Press.

Newey, W. K. and K. D. West (1987, May). A Simple, Positive Semi-Definite, Heteroskedasticity and Autocorrelation Consistent Covariance Matrix. *Econometrica 55*(3), 703–708.

Neyman, A. and S. Sorin (Eds.) (2004). *Stochastic Games and Applications*, Volume 570 of *NATO Science Series C*. New York, NY: Springer.

Ng, A. Y. and M. I. Jordan (2000). Pegasus: a Policy Search Method for Large MDPs and POMDPs. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 406–415.

Ng, A. Y., H. J. Kim, M. I. Jordan, and S. Sastry (2004). Autonomous Helicopter Flight via Reinforcement Learning. In S. Thrun, L. Saul, and B. Schölkopf (Eds.), *Advances in Neural Information Processing Systems 16*, Cambridge, MA. MIT Press.

Ocone, D. L. and I. Karatzas (1991). A Generalized Clark Representation Formula, With Application To Optimal Portfolios. *Stochastics and Stochastics Reports 34*(3), 187–220.

O'Hagan, A. (1978). Curve Fitting and Optimal Design for Prediction. *Journal of the Royal Statistical Society B 40*, 1–42. (With discussion).

Ohlson, J. A. (1995). Earnings, Book Values, and Dividends in Equity Valuation. *Contemporary Accounting Research 11*(2), 661–687.

Ordentlich, E. and T. M. Cover (1998, November). The Cost of Achieving the Best Portfolio in Hindsight. *Mathematics of Operations Research 23*(4), 960–982.

Ormoneit, D. and P. W. Glynn (2001). Kernel-Based Reinforcement Learning in Average-Cost Problems: An Application to Optimal Portfolio Choice. In T. K. Leen, T. G. Dietterich, and V. Tresp (Eds.), *Advances in Neural Information Processing Systems 13*, pp. 1068–1074. MIT Press.

Orr, G. B. and K.-R. Müller (Eds.) (1998). *Neural Networks: Tricks of the Trade*, Volume 1524 of *Lecture Notes in Computer Science*. Springer.

Osorio, M. A., N. Gülpinar, and B. Rustem (2008, January). A General Framework For Multistage Mean-Variance Post-Tax Optimization. *Annals of Operations Research 157*(1), 3–23.

Pan, J. and A. M. Poteshman (2006). The Information in Option Volume for Future Stock Prices. *Review of Financial Studies 19*(3), 871–908.

Phelps, E. S. (1962, October). The Accumulation of Risky Capital: A Sequential Utility Analysis. *Econometrica 30*(4), 729–743.

Philips, T. K. (2003, Fall). Estimating Expected Returns. *Journal of Investing*, 49–57.

Pinheiro, J. C. and D. M. Bates (2000). *Mixed Effects Models in S and S-Plus*. New York, NY: Springer-Verlag.

Pliska, S. R. (1986). A Stochastic Calculus Model of Continuous Trading: Optimal Portfolios. *Mathematics of Operations Research 11*(2), 371–382.

Poitras, G. (1987). "Golden Turtle Tracks": In Search of Unexploited Profits in Gold Spreads. *The Journal of Futures Markets 7*(4), 397–412.

Politis, D. N. and J. P. Romano (1992). A Circular Block-Resampling Procedure for Stationary Data. In R. LePage and L. Billard (Eds.), *Exploring the Limits of Bootstrap*, Wiley Series in Probability and Mathematical Statistics, New York, NY, pp. 263–270. John Wiley & Sons.

Powell, W. B. (2007). *Approximate Dynamic Programming: Solving the Curses of Dimensionality*. Hoboken, NJ: John Wiley & Sons.

Pratt, J. W. (1964). Risk Aversion in the Small and in the Large. *Econometrica 32*, 122–136.

Qian, E. E., R. H. Hua, and E. H. Sorensen (2007). *Quantitative Equity Portfolio Management: Modern Techniques and Applications*. CRC Financial Mathematics Series. Chapman & Hall.

Quiñonero-Candela, J. and C. E. Rasmussen (2005). A Unifying View of Sparse Approximate Gaussian Process Regression. *Journal of Machine Learning Research 6*, 1939–1959.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. ISBN 3-900051-07-0.

Rabiner, L. and B.-H. Juang (1993). *Fundamentals of Speech Recognition*. Prentice Hall.

Rachev, S. T., C. Menn, and F. J. Fabozzi (2005). *Fat-Tailed and Skewed Asset Return Distributions: Implications for Risk Management, Portfolio Selection, and Option Pricing*. Hoboken, NJ: John Wiley & Sons.

Ramsay, J. O. and B. W. Silverman (2005). *Functional Data Analysis* (Second ed.). Springer.

Rasmussen, C. E. and C. K. I. Williams (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rau-Bredow, H. (2004). Value-at-Risk, Expected Shortfall and Marginal Risk Contribution. In G. Szegö (Ed.), *Risk Measures for the 21st Century*, Chichester, pp. 61–68. John Wiley & Sons.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge, UK: Cambridge University Press.

RiskMetrics (1996). RiskMetrics—Technical Document. Technical report, J.P. Morgan, New York, NY. `http://www.riskmetrics.com`.

Robert M. Dammon, C. S. S. and H. H. Zhang (2004, June). Optimal Asset Location and Allocation with Taxable and Tax-Deferred Investing. *Journal of Finance 59*(3), 999–1037.

Rosenberg, B., K. Reid, and R. Lanstein (1985). Persuasive Evidence of Market Inefficiency. *Journal of Portfolio Management 11*, 9–17.

Ross, S. A. (1976, December). The Arbitrage Theory of Capital Asset Pricing. *Journal of Economic Theory 13*(3), 341–360.

Roweis, S. and Z. Ghahramani (1999). A Unifying Review of Linear Gaussian Models. *Neural Computation 11*, 305–345.

Roy, A. D. (1952, July). Safety-First and the Holding of Assets. *Econometrica 20*(3), 431–449.

Rubinstein, M. (2002, June). Markowitz's "Portfolio Selection": A Fifty-Year Retrospective. *Journal of Finance 57*(3), 1041–1045.

Rumelhart, D. E., G. E. Hinton, and R. J. Williams (1986). *Learning Internal Representations by Error Propagation*, Volume Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Chapter 8, pp. 310–362. Cambridge, MA: MIT Press.

Russell, S. J. and P. Norvig (2002). *Artificial Intelligence: A Modern Approach* (Second ed.). Prentice Hall.

Saad, D. (1999). *On-Line Learning in Neural Networks*. Cambridge, UK: Cambridge University Press.

Samuelson, P. A. (1965). Proof that Properly Anticipated Prices Fluctuate Randomly. *Industrial Management Review 6*, 41–49.

Samuelson, P. A. (1969). Lifetime Portfolio Selection by Dynamic Stochastic Programming. *Review of Economics and Statistics 51*(3), 239–246.

Satchell, S. (Ed.) (2007). *Forecasting Expected Returns in the Financial Markets*. London, UK: Academic Press.

Scherer, B. (2002). Portfolio Resampling: Review and Critique. *Financial Analysts Journal 58*, 98–109.

Schroder, M. and C. Skiadas (1999, November). Optimal Consumption and Portfolio Selection with Stochastic Differential Utility. *Journal of Economic Theory 89*(1), 68–126.

Schölkopf, B. and A. J. Smola (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press.

Shadbolt, J. and J. G. Taylor (2002). *Neural Networks and the Financial Markets: Predicting, Combining and Portfolio Optimisation*. Berlin, New York: Springer.

Sharpe, W. F. (1963). A Simplified Model for Portfolio Analysis. *Management Science 9*, 277–293.

Sharpe, W. F. (1964, September). Capital Asset Prices: A Theory of Market Equilibrium Under Conditions of Risk. *Journal of Finance 19*(3), 425–442.

Sharpe, W. F. (1966, January). Mutual Fund Performance. *Journal of Business 39*(1), 119–138.

Sharpe, W. F. (1994). The Sharpe Ratio. *The Journal of Portfolio Management 21*(1), 49–58.

Shawe-Taylor, J. and N. Cristianini (2004). *Kernel Methods for Pattern Analysis*. Cambridge University Press.

Shefrin, H. (2002). *Beyond Greed and Fear: Understanding Behavioral Finance and the Psychology of Investing*. Oxford, UK: Oxford University Press.

Shefrin, H. (2005). *A Behavioral Approach to Asset Pricing*. Burlington, MA: Academic Press.

Shefrin, H. and M. Statman (1985, July). The Disposition to Sell Winners Too Early and Ride Losers Too Long: Theory and Evidence. *Journal of Finance 40*(3), 777–790.

Shefrin, H. and M. Statman (2000, June). Behavioral Portfolio Theory. *Journal of Financial and Quantitative Analysis 35*(2), 127–151.

Shreve, S. E. (2005a). *Stochastic Calculus for Finance I: The Binomial Asset Pricing Model*. New York, NY: Springer.

Shreve, S. E. (2005b). *Stochastic Calculus for Finance II: Continuous-Time Models*. New York, NY: Springer.

Shreve, S. E. and H. M. Soner (1994, August). Optimal Investment and Consumption with Transaction Costs. *Annals of Applied Probability 4*(3), 609–692.

Shumway, R. H. and D. S. Stoffer (1982). An Approach to Time Series Smoothing and Forecasting Using the EM Algorithm. *Journal of Time Series Analysis 3*(4), 253–264.

Si, J., A. G. Barto, W. B. Powell, and D. Wunsch (Eds.) (2004). *Handbook of Learning and Approximate Dynamic Programming*. IEEE Press Series on Computational Intelligence. Wiley–IEEE Press.

Simon, D. P. (1999). The Soybean Crush Spread: Empirical Evidence and Trading Strategies. *The Journal of Futures Markets 19*(3), 271–389.

Skoulakis, G. (2007, March). Dynamic Portfolio Choice with Bayesian Learning. Working Paper, University of Maryland.

Sornette, D. (2003). *Why Stock Markets Crash: Critical Events in Complex Financial Systems*. Princeton University Press.

Sortino, F. A. and L. N. Price (1994, Fall). Performance Measurement in a Downside Risk Framework. *The Journal of Investing*, 59–65.

Sortino, F. A. and R. van der Meer (1991). Downside Risk—Capturing What's at Stake in Investment Situations. *Journal of Portfolio Management 17*(4), 27–31.

Spurgin, R. (1999). A Benchmark for Commodity Trading Advisor Performance. *Journal of Alternative Investments 2*(1), 11–21.

Stein, C. (1956). Inadmissibility of the Usual Estimator for the Mean of Multivariate Normal Distribution. In *Prooceedings of the Third Berkeley Symposium on Mathematical Statistics and Probability*, pp. 197–206.

Stock, J. H. and M. W. Watson (1996). Evidence on Structural Instability in Macroeconomic Time Series Relations. *Journal of Business and Economic Statistics 14*, 11–30.

Stock, J. H. and M. W. Watson (1999). Forecasting Inflation. *Journal of Monetary Economics 44*, 293–335.

Stone, M. (1974). Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society, B 36*(1), 111–147.

Sundararajan, S. and S. S. Keerthi (2001). Predictive Approaches for Choosing Hyperparameters in Gaussian Processes. *Neural Computation 13*(5), 1103–1118.

Sutton, R. S. (1984). *Temporal Credit Assignment in Reinforcement Learning*. Ph. D. thesis, University of Massachusetts, Amherst, MA.

Sutton, R. S. (1988). Learning to Predict by the Method of Temporal Differences. *Machine Learning 3*, 9–44.

Sutton, R. S. and A. G. Barto (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Sutton, R. S., D. McAllester, S. P. Singh, and Y. Mansour (2000). Policy Gradient Methods for Reinforcement Learning with Function Approximation. In S. A. Solla, T. K. Leen, and K.-R. Müller (Eds.), *Advances in Neural Information Processing Systems 12*, Cambridge, MA, pp. 1057–1063. MIT Press.

Taksar, M., M. J. Klass, and D. Assaf (1988, May). A Diffusion Model for Optimal Portfolio Selection in the Presence of Brokerage Fees. *Mathematics of Operations Research 13*(2), 277–294.

Theil, H. and A. Goldberger (1961). On Pure and Mixed Estimation in Economics. *International Economic Review 2*, 65–78.

Tobin, J. (1958, February). Liquidity Preference as a Behavior Towards Risk. *Review of Economic Studies 67*, 65–86.

Tobin, J. (1965). The Theory of Portfolio Selection. In F. H. Hahn and F. P. R. Brechling (Eds.), *The Theory of Interest Rates*, London. Macmillan.

Tresp, V. (2000). A Bayesian Committee Machine. *Neural Computation 12*, 2719–2741.

Tsitsiklis, J. N. and B. V. Roy (1997). An Analysis of Temporal-Difference Learning with Function Approximation. *IEEE Transactions on Automatic Control 42*, 674–690.

Tsitsiklis, J. N. and B. V. Roy (2001). Regression Methods for Pricing Complex American-Style Options. *IEEE Transactions on Neural Networks 12*(4), 694–703.

Tütüncü, R. H. and M. Koenig (November 2004). Robust Asset Allocation. *Annals of Operations Research 132*, 157–187.

U.S. Department of Agriculture (2008). Economic Research Service Data Sets. WWW publication. Available at `http://www.ers.usda.gov/Data/`.

Vapnik, V. N. (1998). *Statistical Learning Theory*. New York, London, Sydney: John Wiley & Sons.

Viceira, L. M. (2001, April). Optimal Portfolio Choice for Long-Horizon Investors with Nontradable Labor Income. *Journal of Finance 56*(2), 433–470.

Vlcek, M. (2006, November). Portfolio Choice with Loss Aversion, Asymmetric Risk-Taking Behavior and Segregation of Riskless Opportunities. Swiss Finance Institute Research Paper No. 27. Available at `http://ssrn.com/abstract=947078`.

Wachter, J. A. (2002). Portfolio and Consumption Decisions Under Mean-Reverting Returns: An Exact Solution for Complete Markets. *Journal of Financial and Quantitative Analysis 37*(1), 63–91.

Wagner, W. H. and M. Edwards (1998). Implementing Investment Strategies: The Art and Science of Investing. In F. J. Fabozzi (Ed.), *Active Equity Portfolio Management*, Hoboken, NJ, pp. 186. John Wiley & Sons.

Wahab, M., R. Cohn, and M. Lashgari (1994). The gold-silver spread: Integration, cointegration, predictability, and ex-ante arbitrage. *The Journal of Futures Markets 14*(6), 709–756.

Wand, M. P. and M. C. Jones (1995). *Kernel Smoothing*. London: Chapman and Hall.

Watkins, C. J. C. H. (1989). *Learning from Delayed Rewards*. Ph. D. thesis, Cambridge University, Cambridge, England.

Watkins, C. J. C. H. and P. Dayan (1992). Q-Learning. *Machine Learning 8*, 279–292.

Weaver, L. and N. Tao (2001). The Optimal Reward Baseline for Gradient-Based Reinforcement Learning. In *Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, pp. 538–545. Morgan Kaufmann.

Weigend, A. S. and N. Gershenfeld (1993). *Time Series Prediction: Forecasting the future and understanding the past*. Reading, MA, USA: Addison-Wesley.

White, H. (2000). A Reality Check for Data Snooping. *Econometrica 68*(5), 1097–1126.

Williams, C. K. I. and C. E. Rasmussen (1996). Gaussian Processes for Regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo (Eds.), *Advances in Neural Information Processing Systems 8*, pp. 514–520. MIT Press.

Williams, J. B. (1938). *The Theory of Investment Value*. Cambridge, MA: Harvard University Press.

Williams, R. J. (1992). Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning. *Machine Learning 8*, 229–256.

Wilmott, P. (2006). *Paul Wilmott on Quantitative Finance* (Second ed.). John Wiley & Sons.

Wolpert, D. H. (1992). Stacked Generalization. *Neural Networks 5*, 241–249.

Wolsey, L. A. and G. L. Nemhauser (1999). *Integer and Combinatorial Optimization*. New York, NY: Wiley-Interscience.

Working, H. (1934). Price Relationships Between May and New-Crop Wheat Future at Chicago Since 1885. *Wheat Studies 10*, 183–228.

Working, H. (1935, oct). Differential Price Behavior as a Subject for Commodity Price Analysis. *Econometrica 3*(4), 416–427.

Working, H. (1949, dec). The Theory of Price of Storage. *The American Economic Review 39*(6), 1254–1262.

Xia, Y. (2001, February). Learning about Predictability: The Effects of Parameter Uncertainty on Dynamic Asset Allocation. *Journal of Finance 56*(1), 205–246.

Yu, L.-Y., X.-D. Ji, and S.-Y. Wang (2003). Stochastic Programming Models in Financial Optimization: A Survey. *Advanced Modeling and Optimization 5*(1), 1–26.

Zeileis, A. (2004). Econometric Computing with HC and HAC Covariance Matrix Estimators. *Journal of Statistical Software 11*(10), 1–17.

Zellner, Z. A. and V. K. Chetty (1965). Prediction and Decision Problems in Regression Models from the Bayesian Point of View. *Journal of the American Statistical Association 60*, 608–615.

Zenios, S. A. (Ed.) (1993). *Financial Optimization*. Cambridge, UK: Cambridge University Press.

Zenios, S. A. and W. T. Ziemba (Eds.) (2006). *Handbook of Asset and Liability Management, Volume 1: Theory and Methodology*. Handbooks in Finance. Amsterdam, Netherlands: North Holland.

Zhou, W.-X. and D. Sornette (2007). Self-Organizing Ising Model of Financial Markets. *European Physical Journal B 55*, 175–181.

Zhou, X. Y. (2000, December). Continuous-Time Mean-Variance Portfolio Selection: A Stochastic LQ Framework. *Applied Mathematics and Optimization 42*(1), 19–33.

Zimmermann, H.-G., L. Bertolini, R. Grothmann, A. M. Schäfer, and C. Tietz (2006). A Technical Trading Indicator Based on Dynamical Consistent Neural Networks. In *ICANN (2)*, pp. 654–663.

Zimmermann, H.-G., R. Grothmann, A. M. Schäfer, and C. Tietz (2006). Modeling Large Dynamical Systems with Dynamical Consistent Neural Networks. In S. Haykin, J. C. Príncipe, T. J. Sejnowski, and J. McWhirter (Eds.), *New Directions in Statistical Signal Processing: From Systems to Brains*, Cambridge, MA, pp. 203–242. MIT Press.

Zimmermann, H.-G., R. Neuneier, and R. Grothmann (2001). Active Portfolio-Management based on Error Correction Neural Networks. In T. G. Dietterich, S. Becker, and Z. Ghahramani (Eds.), *Advances in Neural Information Processing Systems 14*, pp. 1465–1472. MIT Press.

Author Index

Abu-Mostafa, Yaser S. 89, 167
Alexander, Gordon J. 17
Andrews, Donald W. K. 193, 199
Arrow, Kenneth J. 15
Artzner, Philippe 18
Assaf, David 57
Atiya, Amir F. 89
Atkinson, Colin 58
Aït-Sahalia, Yacine 31, 54, 63

Bagnell, J. Andrew 86, 136
Banz, R. W. 26
Baptista, Alexandre M. 17
Barberis, Nicholas 33, 55
Barra 26
Barto, Andrew G. 77, 82, 85, 88, 134
Basu, Sanjoy 26
Bates, Douglas M. 186
Bauer, David F. 161
Bawa, V.S. 33
Beasley, J. E. 24
Beitelspacher, Josh 88
Bell, Philip W. 30
Bellman, Richard Ernest 39, 57, 81, 82, 121, 140
Ben-Tal, A. 36
Bengio, Yoshua 60, 89, 90, 121, 122, 137, 169, 243
Benveniste, A. 73, 84
Bera, Anil K. 186
Berkelaar, Arjan B. 57
Bernoulli, Daniel 2
Bertolini, Lorenzo 61
Bertsekas, Dimitri P. 19, 39, 73, 77, 79, 82, 85, 86, 121, 122, 131, 135, 140, 168, 251
Bertsimas, Dimitris 37
Best, M. J. 31
Bevan, Andrew 35
Bienenstock, E. 73
Birge, John R. 65, 66
Bishop, Christopher M. 4, 69, 73, 168, 171, 243, 292
Black, Fisher 33, 112, 116, 117
Blum, A. 65
Bodie, Z. 19
Bollerslev, T. 90
Booch, Grady 111
Bornholdt, Stefan 133
Boser, Bernhard E. 72
Bouchaud, Jean-Philippe 27
Box, George Edward Pelham 194, 199
Boyd, Stephen 37
Brandt, Michael W. 28, 31, 32, 44, 47, 54, 55, 57, 59, 61–63
Breiman, Leo 93, 112, 192
Brennan, M. J. 28
Brown, Keith C. 130
Brown, Stephen J. 33
Burges, Christopher 162
Butterworth, Darren 247

Campbell, John Y. 15, 25, 28, 56, 90
Candela, Joaquin Quiñonero 245
Carhart, Mark M. 26
Chang, Hyeong Soo 88
Chang, T.-J. 24
Chapados, Nicolas 14, 17, 60, 89, 90, 121, 122, 192, 243
Chetty, V. K. 33
Choey, M. 60
Chopra, Vijay K. 31
Chou, Chih-Hsien 247
Chow, George 17
Christoffersen, Peter F. 162, 207
Chu, Wei 162
Cizeau, Pierre 27
Claus, James 30
Clements, Michael P. 90, 95, 264
Cohn, Richard 247
Collins, Art 197
Conallen, Jim 111
Consigli, Giorgio 18

Constantinides, George M. 57
Cormen, Thomas H. 105, 152
Cover, Thomas M. 64
Cox, John C. 49, 51, 54
Cressie, Noel A. C. 244
Crichfield, Timothy 30
Cristianini, Nello 260
Cvitanić, Jakša 44, 54, 58, 59, 130

Dammon, Robert M. 59
Dantzig, George B. 65, 68
Davis, M. H. A. 57
Dawid, A. Philip 91
Dayan, Peter 63, 80
De Bondt, Werner F. M. 26
Delbaen, Freddy 18
DeMiguel, Victor 37, 59
Dempster, A. P. 103
Dempster, M. A. H. 68
Detemple, Jérôme B. 54
Diebold, Francis X. 162, 207, 265
Dodd, David L. 3
Doursat, R. 73
Dreyfus, Stuart E. 81, 82, 138, 140
Duffie, Darrell 44
Dunis, Christian L. 60, 61, 116, 247, 254
Dutt, Hans R. 247
Dyckman, Thomas 30

Eastham, Jerome F. 58
Eber, Jean-Marc 18
Edwards, Edgar O. 30
Edwards, Mark 21
Elton, Edwin J. 59
Engel, Michael W. 111
Engle, R. 90
Eppstein, David 138
Epstein, Larry G. 56
Erb, Claude B. 114
Estrada, Javier 16
Evans, Ben 60, 61, 247, 254

Fabozzi, Frank J. 10, 11, 18, 19, 21, 27, 33, 35, 36, 112
Fager, Jason 88
Fama, Eugene F. 26, 28, 90
Farinelli, Simone 18
Fenton, John 247
Fisher, Irving 3
Fisher, Kenneth L. 200
Focardi, Sergio M. 10, 11, 19, 21, 27, 33, 35, 36, 112
Fox, Emily B. 293
Frasconi, Paolo 137, 243
French, Kenneth R. 26, 28, 90
Freund, Yoav 93
Friedman, Jerome 69, 70, 75, 90
Friesen, Geoffrey 30
Frost, P. A. 31, 32
Fu, Michael C. 88
Fujiwara, Yoshi 133
Fung, William 114

Gallmeyer, Michael F. 59
Garcia, René 54
Garlappi, Lorenzo 37
Geman, S. 73
Germano, M. 68
Gershenfeld, Neil 60
Ghahramani, Zoubin 100, 103, 162
Ghosn, Joumana 60
Gingras, François 90
Girard, Agathe 245
Girma, Paul Berhanu 246
Givoly, Dan 30
Glantz, Morton 21, 40
Glicksberg, I. 57
Glynn, Peter W. 63, 87
Goldberg, Paul W. 292
Goldberger, A. 35
Goldfarb, Donald 36
Gordon, Myron 29
Goukasian, Levon 54
Goyal, Amit 55
Graham, Benjamin 3
Grauer, R. 31
Greene, William H. 35, 72, 93, 193, 199, 296
Grinold, Richard C. 20, 22, 121, 268
Gross, O. 57
Grothendieck, Gabor 199
Grothmann, Ralph 61
Gruber, Martin J. 59
Guyon, Isabelle M. 72
Gülpinar, Nalan 59

Halldórsson, Bjarni Vilhjálmur 37
Hamilton, James D. 28, 243, 263
Hansen, Peter R. 89
Harlow, W. V. 130
Harrison, M. 49
Hart, P. E. 151
Hartmann, Alexander K. 131

Harvey, Campbell R. 114
Hassibi, Babak 100
Hastie, Trevor 69, 70, 75, 90
Hastings, Kevin J. 58
Hautsch, Nikolaus 96
Heath, David 18
Helmbold, David P. 112
Hendry, David F. 90, 95, 264
Henriques, Greg 88
Hens, Thorsten 63
Herbster, Mark 112, 118
Hess, Dieter 96
Hinton, Geoffrey E. 60, 73, 103, 169
Hoerl, A. E. 116
Holmes, Phil 247
Hornik, Kurt 73
Houston, Kelli A. 111
Hsieh, David A. 114
Hsu, Jason 196
Hua, Ronald H. 19, 27
Huang, Chi-Fu 49, 51, 54
Hull, John C. 114, 244
Hunter, J. S. 194
Hunter, W. G. 194
Härdle, W. 62

Infanger, Gerd 68
Itô, Kiyoshi 298
Iyengar, Garud 36

Jaakkola, Tommi 87
Jagannathan, R. 32
James, W. 32
Jarque, Carlos M. 186
Jegadeesh, Narasimhan 26, 114
Ji, Xiao-Dong 68
Jiménez, Víctor M. 138, 142
Jin, Hanqing 16
Jobson, J. Dave 31, 32, 192
Jones, M. Chris 149
Jordan, Michael I. 87, 88, 136, 293
Jorion, Philippe 22, 32
Juang, Biing-Hwang 135

Kahn, Ronald N. 20, 22, 121, 268
Kahneman, Daniel 57
Kailath, Thomas 100
Kaizoji, Taisei 133
Kakade, Sham 86, 136
Kalaba, Robert 140
Kalai, Adam 65
Kallberg, Jerry G. 15
Kalman, Rudolf E. 100
Kandel, Shmuel 28, 33, 55
Kane, A. 19
Kaniel, Ron 59
Karatzas, Ioannis 49, 54, 58
Kaufman, P. J. 63
Kearns, Michael J. 136
Keerthi, S. S. 251
Kellerer, Hans 24
Kelton, W. David 88
Kennard, R. W. 116
Kim, H. Jin 88
Kim, Min Kyoung 246
Kim, Tong S. 48
Kirkpatrick, Scott 131
Kissell, Robert 21, 40
Kivinen, Jyrki 118
Klass, Michael J. 57
Klein, R.W. 33
Koenig, M. 36
Kolm, Petter N. 10, 11, 19, 21, 27, 33, 35, 36, 112
Konda, Vijay R. 87
Koo, Hyeng Keun 56
Korkie, Bob M. 31, 192
Korn, Ralf 42
Kouwenberg, Roy 57
Kreps, D. 49
Kritzman, Mark 17
Krokhmal, Pavlo 18

Lagnado, R. 28
Lagoudakis, Michail G. 136
Lahiri, Soumendra Nath 191, 193
Laird, N. M. 103
Lakonishok, Josef 26, 30
Laloux, Laurent 27
Lamblin, Pascal 169
Langford, John 136, 137
Lanstein, Ronald 26
Larochelle, Hugo 169
Lashgari, Malek 247
Law, Averill M. 88
Laws, Jason 60, 61, 247, 254
Lazrak, Ali 44, 130
Ledoit, Olivier 32, 178, 187, 192, 193
Lehoczky, John P. 49
Leippold, Markus 44
Leiserson, Charles E. 105, 152
Leland, Hayne E. 58

Leuthold, Raymond M. 246
Levesque, Hector 131
Levy, Haim 15
Li, Duan 42
Li, Ling 162
Li, Ping 162
Liao, Yuansong 63
Lin, Hsuan-Tien 162
Lintner, John 25
Litterman, Robert 33, 112, 116, 117
Liu, Hong 58
Liu, Shi-Miin 247
Ljung, Greta M. 199
Lo, Andrew W. 25, 28, 89, 90, 95
Longstaff, Francis A. 55
Louveaux, François 65, 66
Luenberger, David G. 19, 52
Lunde, Asger 89

Ma, T. 32
MacKay, David J. C. 190, 252, 254
MacKinlay, A. Craig 25, 28, 89, 90, 95
Magdon-Ismail, Malik 89
Magill, Michael J. P. 57
Maksimchuk, Robert A. 111
Malevergne, Yannick 18, 27
Mansini, Renata 24
Mansour, Yishay 87, 136
Marbach, Peter 87
Marcus, A. J. 19
Marcus, Jiaqiao Hu Steven I. 88
Mariano, Roberto S. 162, 207, 265
Markowitz, Harry M. 1, 2, 9, 14–16, 36, 38, 112, 116, 271
Marzal, Andrés 138, 142
Matheron, G. 244
Mathieson, Mark 162, 165, 166
McAllester, David 87
McCullagh, Peter 162, 164
McCulloch, Charles E. 186
McGovern, Amy 88
McNelis, Paul D. 60
Meade, N. 24
Medova, E. A. 68
Memmel, Christoph 192
Menn, Christian 18
Merton, Robert C. 3, 11, 13, 15, 16, 25, 38, 44, 45, 47–49, 51, 54–56, 121, 323
Metivier, M. 73, 84
Miao, Jia 116
Michaud, Richard O. 31, 35
Miffre, Joëlle 114, 115
Mihatsch, Oliver 63
Mitchell, David 131
Mittnik, Stefan 17
Monahan, J. Christopher 193
Montier, James 57
Moody, John 63, 122, 147
Morton, Andrew J. 58
Mossin, Jan 3, 15, 25, 38, 40, 121
Murray-Smith, Roderick 245
Muthuraman, Kumar 58
Mézard, Marc 133

Nadeau, Claude 90
Nawrocki, D. 16
Neal, Radford M. 103, 244, 254
Nelder, John A. 162, 164
Nemhauser, George L. 19, 23
Nemirovski, A. 36
Neuhaus, John M. 186
Neuneier, Ralph 61, 63
Newey, Whidney K. 29, 199
Ng, Andrew Y. 86, 88, 136
Ng, Wan-Lung 42
Nilsson, N. J. 151
Norman, A. R. 57
Norvig, Peter 151

Ocone, Daniel L. 54
O'Hagan, Anthony 244
Ohlson, James A. 30
Omberg, Edward 48
Ordentlich, Erik 64
Ormoneit, Dirk 63
Osindero, Simon 169
Osorio, Maria A. 59

Pachamanova, Dessislava A. 11, 36, 37, 112
Palmquist, Jonas 18
Pan, Jun 31
Parisi, Giorgio 133
Parr, Ronald 136
Paulson, Albert S. 246
Phelps, Edmund S. 38
Philips, Thomas K. 30
Pinheiro, José C. 186
Pliska, Stanley R. 49, 58
Poitras, Geoffrey 247
Politis, Dimitris N. 193
Popovici, Dan 169

Post, Thierry 57
Poteshman, Allen M. 31
Potters, Marc 27
Powell, Warren B. 77, 290
Pratt, John W. 15
Price, Lee N. 121
Priouret, P. 73, 84

Qian, Edward E. 19, 27
Quiñonero-Candela, Joaquin 260, 280

R Development Core Team 199
Rabiner, Lawrence 135
Rachev, Svetlozar T. 17, 18
Rallis, Georgios 114, 115
Ramsay, James O. 243
Raphael, B. 151
Rasmussen, Carl Edward 72, 170, 244, 245, 248, 250, 252, 260, 280
Ratti, V. 32
Rau-Bredow, Hans 18
Reid, Kenneth 26
Rietbergen, M. I. 68
Rindisbacher, Marcel 54
Ripley, B. D. 69
RiskMetrics 17, 60, 116, 147
Rivest, Ronald L. 105, 152
Robert M. Dammon, Chester S. Spatt 59
Romano, Joseph P. 193
Rosenberg, Barr 26
Ross, Stephen A. 25
Rossello, Damiano 18
Roweis, Sam 100
Roy, Andrew D. 16
Roy, Benjamin Van 86
Rubin, D. B. 103
Rubinstein, Mark 2
Rumelhart, David E. 60, 73
Russell, Stuart J. 151
Rustem, Berç 59

Saad, David 103, 113
Saffell, Matthew 63, 122, 147
Samuelson, Paul A. 3, 16, 28, 38, 40, 55, 121
Sandrini, F. 68
Santa-Clara, Pedro 54, 55, 61
Sastry, Shankar 88
Savarino, J. E. 31, 32
Sayed, Ali H. 100
Schapire, Robert E. 93, 112
Scherer, Bernd 35
Schneider, Jeff 86, 136
Schroder, Mark 57
Schwartz, Eduardo S. 17, 28, 55
Schäfer, Anton Maximilian 61
Schölkopf, Bernhard 72
Scrowston, M. 68
Searle, Shayle R. 186
Selman, Bart 131
Shadbolt, Jimmy 60, 89
Sharaiha, Y. M. 24
Sharpe, William F. 13, 17, 25, 27, 119, 121, 123, 296
Shawe-Taylor, John 260
Shefrin, Hersh 57
Shiller, Robert J. 28
Shleifer, Andrei 26
Shreve, Steven E. 49, 58, 298
Shumway, R. H. 103
Silverman, Bernard W. 243
Simard, Patrice 137, 243
Simon, David P. 246
Singer, Yoram 112
Singh, Satinder P. 87
Skiadas, Costis 57
Skoulakis, Georgios 55
Smith, Jonathan D. 247
Smola, Alexander J. 72
Soner, H. M. 58
Sorensen, Eric H. 19, 27
Sornette, Didier 18, 27, 133
Sortino, Frank A. 121
Spatt, Chester S. 59
Speranza, Maria Grazia 24
Spurgin, Richard 114
Stambaugh, Robert F. 28, 33, 55
Starks, Laura T. 130
Statman, Meir 57
Stein, Charles 32
Stein, Clifford 105, 152
Stinchcombe, Maxwell B. 73
Stock, James H. 90, 91
Stoffer, D. S. 103
Stone, M. 76, 90, 251
Stroud, Jonathan R. 55
Sudderth, Erik B. 293
Sundararajan, S. 251
Sutton, Richard S. 77, 82, 84, 85, 87, 88, 134

Taksar, Michael 57
Tao, Nigel 88
Tay, Anthony S. 162, 207
Taylor, John G. 60, 89
Teh, Yee-Whye 169
Thaler, Richard 26
Theil, H. 35
Thomas, Jacob 30
Thomas, Joy A. 64
Tibshirani, Robert 69, 70, 75, 90
Tietz, Christoph 61
Titman, Sheridan 26, 114
Tobiletti, Luisa 18
Tobin, James 13, 38
Tompaidis, Stathis 59
Trautmann, Siegfried 42
Tresp, Volker 280
Trojani, Fabio 44
Tse, Yiu Kuen 162, 207
Tsitsiklis, John N. 77, 82, 85–87, 135
Tversky, Amos 57
Tütüncü, Reha H. 36, 37

Uppal, Raman 37, 59
Uryasev, Stanislav 18
U.S. Department of Agriculture 262
Usmen, Nilufer 36

Valkanov, Rossen 61
van der Meer, R. 121
Vandenberghe, Lieven 37
Vanini, Paolo 44
Vapnik, Vladimir N. 59, 71, 72, 75, 280
Vempala, Santosh 65
Viceira, Luis M. 15, 56
Villaverde, M. 68
Virasoro, Miguel Angel 133
Vishny, Robert W. 26
Vlcek, Martin 57

Wachter, Jessica A. 54
Wagner, Wayne H. 21
Wahab, Mahmoud 247
Wand, Matt P. 149
Wang, George H.K. 247
Wang, Shou-Yang 68
Wang, Tan 44, 130
Warmuth, Manfred K. 112, 118
Watkins, Christopher J. C. H. 63, 80
Watson, Mark W. 90, 91
Weaver, Lex 88
Weigend, Andreas S. 60
Weigt, Martin 131
Weller, Paul A. 30
West, Kenneth D. 29, 199
White, Halbert 73, 89, 95
Williams, Christopher K. I. 72, 170, 244, 245, 248, 250, 252, 260, 292
Williams, John Burr 3, 29
Williams, Ronald J. 60, 73, 87
Willsky, Alan S. 293
Wilmott, Paul 54, 58
Winkelmann, Kurt 35
Wolf, Michael 32, 178, 187, 192, 193
Wolpert, D. H. 93, 111
Wolsey, Laurence A. 19, 23
Working, Holbrook 246
Wu, Lizhong 63
Wu, Qiang 162
Wöhrmann, Peter 63

Xia, Yihong 55

Ye, Yinyu 19, 52
Young, Bobbi J. 111
Yu, Li-Yong 68

Zadrozny, Bianca 136, 137
Zapatero, Fernando 54
Zeileis, Achim 199
Zellner, Z. A. 33
Zha, Haining 58
Zhang, Harold H. 59
Zhou, W.-X. 133
Zhou, Xun Yu 16, 42
Ziemba, William T. 15, 31
Zimmermann, Hans-Georg 61
Zin, Stanley E. 56

Subject Index

ϵ-greedy policy, 82
1/N allocation rule, 37

active risk, 22
    parametric portfolio policies, 62
actor–critic algorithm, 87
additive utility function, 121
allocation
    asset, 9
analysis of variance, 194
annual sharpe ratio
    performance measure, 179
annualized return
    performance measure, 179
anomalies
    asset prices, 26
ANOVA, *see* analysis of variance
approximate dynamic programming, 77, 136
APT, *see* arbitrage pricing theory
arbitrage pricing theory, 25
ARD, *see* automatic relevance determination
ARIMA, *see* autoregressive integrated moving average
artificial neural network, 73
asset pricing anomalies, 26
asset return
    forecasting, 24–31
asset returns
    cross-section, 25
    expected, 10
asset–liability management, 68
automatic relevance determination, 253
autoregressive integrated moving average, 243
autoregressive model, 263
average continuously compounded return, 49
avg annual return
    performance measure, 179
avg annual stddev
    performance measure, 179
avg monthly return
    performance measure, 179
avg monthly stddev
    performance measure, 179

backcasting
    European variables, 208
bagging
    learning algorithm, 93
Bayes' theorem, 249
Bayesian forecasting in portfolio choice, 32
beam search, 151
bear market, 177
behavioral finance, 57
behavioral portfolio theory, 57
Bellman equation, 78, 134
    generalization for K best paths, 139
    in continuous-time portfolio choice, 45
    in discrete-time portfolio choice, 39
benchmark
    role in performance evaluation, 22
Benders decomposition, 68
best month
    performance measure, 179
bias, 74
bias–variance trade-off, 73
Black–Litterman model, 33, 116
boosting
    learning algorithm, 93
bootstrap test
    Sharpe ratio difference, 192
Brownian motion, 44
budget constraint, 15
    in multiperiod portfolio choice, 39
bull market, 177

CAC 40, 131, 154
calendar spread, 247
Calmar ratio, 121
Canada, 172
    fundamental variables, 202
capacity control, 76
capital asset pricing model, 12, 25, 26, 33
    continuous-time, 47
capital market line, 12
CAPM, *see* capital asset pricing model
certainty equivalent return, 37

circular block bootstrap, 193
clean surplus relationship, 29
commodity futures
  calendar spreads, 247
  evidence of forecastability, 246
  price spreads, 244
  review, 246–247
  spread seasonalities, 246
  theory of price of storage, 246
commodity trading advisor, 114
  fundamental model, 116
  technical model, 115
composition
  learning modules, 93
ComputeOutput, 109
conditional expectation, 70
conditional value-at-risk
  risk measure, 18
conjugate gradient optimization, 130, 168, 213, 251, 253, 261
constraint
  cardinality, 23
  factor exposure, 22
  maximum holdings, 21
  maximum tracking error, 22
  no short-sales, 19
  nonnegativity, 19
  portfolio choice, 19–24
  round lot, 23
  transaction costs, 20
  transaction size, 23
  turnover, 20
consumption, 37, 55, 121
continuous-time Bellman equation, 45
continuous-time multiperiod portfolio choice, 44–55
covariance function, 171, 250
  automatic relevance determination, 253
  engineering, 260
  rational quadratic, 250
  squared exponential, 250
covariance matrix
  estimation, 27
  from a Gaussian process, 249
  in commodity trading advisor, 116
  in tangency portfolio, 296
  resulting from Gaussian process, 245
  returns, 10
Cox–Huang equivalence theorem, 51
Cox–Huang optimal weights, 53
cross-validation, 76
CRRA (Constant Relative Risk Aversion), 15
CTA, *see* commodity trading advisor
curse of dimensionality, 4, 62, 121
CVaR, *see* conditional value-at-risk

data snooping biases, 89, 94
DAX 30, 131, 154
decision theory, 32
deep network, 169
delta method
  asymptotic covariance matrix, 193
deterministic policy, 78
deterministic shortest path problem, 122
Diebold-Mariano test, *see* test, Diebold-Mariano
diffusion process, 44
direct maximization of financial criteria, 60
dirty flag
  temporal learning, 108
discount factor, 77
discrete-time Bellman equation, 39
discrete-time multiperiod portfolio choice, 38–44
disposition effect, 57
distribution
  Gaussian, *see* distribution, normal
  normal, 299
  over functions, *see* Gaussian process
diversification, 1, 12
dividend discount model, 29
DotProductOrderEnumeration, 153
Dow Jones Eurostoxx 50, 172
DP, *see* dynamic programming
drawdown duration
  performance measure, 179
drawdown from
  performance measure, 179
drawdown until
  performance measure, 179
dynamic programming, 4, 77, 121
  claimed unsuitability for multiperiod portfolio choice, 38
  non-additive cost structure, 147
dynamical system
  learning primitives, 98

econometric issues in forecasting, 31–37
economic data
  historical revisions, 95
  vintage data, 97
efficient frontier, 1, 9, 123, 297
  mean-semivariance, 16
efficient market hypothesis, 27
efficient portfolio, 11
equilibrium model
  relation to Black–Litterman model, 33
Europe, 172
  backcasting, 208

fundamental variables, 204
Eurostoxx 50, 172
daily model, 208
monthly model, 207
evidence
Bayesian, 251
excess kurtosis
performance measure, 179
expected return, 1
in commodity trading advisor, 116
in tangency portfolio, 296
expected return estimation, 27
fundamental analysis, 29
expected utility, 14
expected utility maximization, 133
Bayesian portfolio optimization, 33
non-additive utility, 134
exploration–exploitation trade-off, 82
exponentiated-gradient mixture, 117

factor exposure, 22
factor models, 25–29
commercial, 26
covariance matrix estimation, 27
definition, 25
expected return estimation, 27
Fama–French factors, 26
FDA, *see* functional data analysis
feature, 81
financial model
classification neural network, 171
financial neural network, 171
Gaussian process, 170
linear regression, 170
long-only, 170
financial network
financial objective, 166
overall objective, 168
prefitting, 168
relationship to deep networks, 169
training and regularization, 165
financial neural network, 162
architecture, 164
fine-tuning
gradient-based, 154
forecast stability, 31–37
impact of uncertainty , 31
forecastability
asset returns, 27
forecasting, *see* time-series forecasting
complete future trajectory, 244
seasonal effects, 244
forecasting model
asset returns, 24–31
forecasting models, 9
fraction months up
performance measure, 179
friction, 3
continous-time, 58
function class
kernel machines, 72
linear, 71
neural network, 73
functional data analysis, 243
fundamental variable
Canada, 202
definition, 197
Europe, 204
United States, 199
futures, *see* commodity futures

GARCH, *see* Generalized Autoregressive Conditional Heteroscedasticity
Gaussian distribution
inference, 299
Gaussian process, 72, 244
connection with neural networks, 244
definition, 248
overfitting, 252
prior, 248
regression, 247–254
subsampling, 260
two-step training, 252
generalization error, 70
decomposition, 75
optimistic bias in estimation, 95
unbiased estimator, 71
Generalized Autoregressive Conditional Heteroscedasticity, 90
geometric Brownian motion, 44, 298
globally minimum-variance portfolio, 12
Gordon growth model, 29
gradient-based fine-tuning, 154
greedy policy, 78
growth-optimum portfolio, 49

HARA, *see* hyperbolic absolute risk aversion
hedge fund, 3
hedging demands
component of continuous-time solution, 46
high-capacity model
bad financial performance, 175
historical data, 4
honestly significant differences, 196
horizon effects
in multiperiod portfolio choice, 42
in Sharpe ratio maximization, 128
hyper-mixture, 118
hyperbolic absolute risk aversion, 48, 54, 56, 57

hyperparameter
  list used in experiments, 212
  optimization
    Gaussian processes, 251
  parameterization, 253
hyperparameter selection, 172
hyperprior, 251

information ratio, 121
  use in trading rule, 268
information theory, 64
instability
  model and hyperparameters, 174
institutional investor, 3, 130
interaction effect
  in analysis of variance, 194
intermediate consumption, 37, 55
intertemporal expected utility
    maximization, 38
investor learning, 55
Ising model
  relationship to finance, 131
Itô vector process, 44
Itô's lemma, 298

James-Stein shrinkage estimator, 32

K best paths, 122, 135
  application to portfolio optimization, 142
  choosing a good K, 146
  efficient action enumeration, 152
  enumeration algorithm, 138–142
  experimental ability to optimize, 154–161
  experimental results, 173–189
  improving search efficiency, 150
  limitations, 137
  search parameters, 157
  use in speech recognition, 135
Kalman filter, 100
Karush-Kuhn-Tucker conditions, 52
kernel machine, 72
Kriging, 244

labor income, 37, 55
Lagrange multiplier, 295
lagrange multiplier
  minimum-variance formulation, 11
learning
  of investors in portfolio choice, 55
learning algorithms, 69
learning module composition, 93
leverage, 12, 30, 124
leverage scale
  in Sharpe ratio criterion, 127
linear functions, 71
linear mixed-effects model, 186
logarithmic utility, 15
logit, 163
loss function, 69
  regression, 70

machine learning, 69
  role in portfolio management, 4
main effect
  in analysis of variance, 194
marginal likelihood, 76, 251
Markov chain, 78
martingale method for portfolio choice
  formulation, 49–54
  implementation, 54
maximum drawdown
  performance measure, 179
mean–variance
  multiperiod criterion, 42
  relation to single-period portfolio choice, 9
  versus information-ratio trading rule, 271
mean-VaR optimization, 60
mixed estimation, 35
mixed-integer programming, 19, 23
model capacity, 75
modern portfolio theory, 9
momentum
  technical trading rule, 115
Monte Carlo
  approximation of the expected Sharpe ratio, 125
  use in PEGASUS algorithm, 136
Monte Carlo importance sampling, 68
Monte Carlo policy evaluation, 83
multi-layer perceptron, 73
multiperiod
  portfolio choice formulation, 3
multiperiod portfolio choice, 37–59, 121
multiple comparisons
  in statistical tests, 196
mutual fund, 3
myopic
  portfolio choice, 3, 9
myopic policy
  optimality, 49
myopic portfolio
  component of continuous-time solution, 46

neural network, 73
neurodynamic programming, 77
NextPath, 141
noise, 74

consequences for comparing models, 171
non-additive utility function, 134
non-allocation, 63
non-stationarity
data distribution, 90
nonparametric portfolio weights, 62
normal distribution
inference, 299

odds
parameterization of probabilities, 163
operation time, *see* time-series forecasting, augmented functional representation
optimal policy
comparison between power utility and Sharpe ratio, 129
optimal weights
structure in single-period formulation, 12
ordinal regression
proportional odds model, 163
use in portfolio allocation, 161–169
ordinary least squares, 71
overfitting, 89

parametric portfolio policies, 61
partially observed Markov decision process
use of K best paths, 137
partitioned matrix
inverse, 296
PEGASUS algorithm, 87, 136
performance bound
approximate value iteration, 82
policy gradient, 87
phase transition
in combinatorial optimization, 131
policy evaluation, 79
Monte Carlo, 83
temporal differences, 84
policy gradient
comparison with value-based approximation, 88
policy gradient algorithm, 86
policy improvement, 79, 85
policy iteration, 79
policy learning, 81, 86
population and evolutionary approaches, 88
POMDP, *see* partially observed Markov decision process
portfolio, 2
portfolio choice, 1
constraints, 19–24
direct and alternative methods, 59–68
information-theoretic approaches, 64
minimum-variance formulation, 11
multiperiod, 37–59
continuous-time, 44–55
discrete-time, 38–44
extensions, 55–59
first-order optimality conditions, 40
numerical example, 42
power utility solution, 40
single-period, 9–37
supervised learning, 60
utility maximization formulation, 14
portfolio resampling, 35
positive-definite matrix
notation, 7
predictive distribution
Gaussian processes, 249
preferences
non-standard, 56
prefitting, 168
importance for financial performance, 176
preprocessing
input and target variables, 259
prequential analysis, 91
price of risk, 47
prior distribution
asset returns, 33
probability distribution, 69
prospect theory, 57

Q-value function, 78
quadratic form
minimization under constraints, 295
quadratic loss, 70
relation to shrinkage methods, 32
quadratic programming, 19
quadratic utility, 14
comparison to alternative utility functions, 15

random variables, 69
rational quadratic, 171
realized performance, 4
recourse
in stochastic programs, 65
recurrent neural networks
portfolio choice, 60
recursive enumeration algorithm
data structures, 141
for K best paths, 138
RecursivelyEnumerateKShortestPaths, 141
reduction
in machine learning problems, 136
regression, 70
regularization

role of constraints in portfolio allocation, 19
reinforcement learning, 77
in trading systems, 63
required return
on equity, 29
rescoring, 134
impact of K on Sharpe ratio, 154
residual income valuation, 29
return
asset, definition, 6
dividend-paying asset, 6
revision
economic data, 95
impact on market volatility, 96
risk, 1, 2
risk aversion, 14
risk factor, 22
risk measure, 16–18
risk premium, 48
risk-free asset, 12
notational convention, 7
risky asset, 15
robust portfolio allocation, 36
rollover policy, 255
Roy safety first
risk measure, 16
Russell 1000, 131, 154

S&P 500, 131, 154, 172
daily model, 202
monthly model, 200
S&P/TSX Composite, 172
saddle-point problem, 37
safety first
risk measure, 16
semivariance, 16
sequential decision-making, 89
sequential learning
correctness issues, 94
sequential validation, 5, 77, 90, 262
training set construction, 92
SequentialValidation, 105
shadow price of wealth, 52
Sharpe ratio, 13, 15, 121
bootstrap inference, 192
difference significance test, 178
difficulty to optimize, 130–133
empirical, 123
leverage scale, 127–128
optimal policy, 128–130
optimization, 123–133
out of sample, 37
regularization, 125–127
tournament comparison, 178
versus tangency portfolio, 296
shrinkage estimation, 32
simulated out-of-sample, 91
single-period portfolio choice, 9–37
skewness
performance measure, 179
return distribution, effect on semivariance, 16
Sortino ratio, 121
source utility, 135
impact on performance, 155
incremental Sharpe ratio, 146
MinCost, 144
bound on value function, 144
standard error, 193
adjustment in time-series regressions, 198
state variable, 103
stationary policy, 78
statistical error maximizer, 31
statistical learning, 69
stochastic differential equation, 298
stochastic gradient optimization, 129
stochastic programming
cultural gap, 68
use in portfolio choice, 65
stochastic shortest-path problem, 77
structural break
economic data, 90
structured estimator
relation to shrinkage methods, 32
studentized test statistic, 193
subadditivity
desideratum for risk measures, 18
supervised learning, 69
support vector machine, 72

tangency portfolio, 12, 296
versus Sharpe ratio, 123
target time, *see* time-series forecasting, augmented functional representation
target utility, 135
Sharpe ratio, 145
taxes
continuous-time, 58
technical variable
calendar events, 198
definition, 197
list of variables used in experiments, 197
temporal difference
non-tabular representations, 85
tabular representations, 84
temporal learning algorithm
initial state, 100
output computation, 99

state transition, 99
state variable, 103
training, 100
temporal learning network, 104
domain analysis, 110
terminal state, 77
terminal wealth, 121
test
cross-covariance-corrected Diebold-Mariano, 267
Diebold-Mariano, 265
equality of means, *see* analysis of variance
Sharpe ratio difference, 192
test set, 70
time-series forecasting, 243
augmented functional representation, 257–259
functional representation, 254–261
performance difference significance test, 265–267
time-series regression
adjustment of standard errors, 198
TOPIX, 131
topological sort
temporal learning network, 104
tracking error, 22
Train, 107
training set, 70
transaction cost, 3, 20, 37, 131
continuous-time, 57
explicit, 21
impact in Sharpe ratio optimization, 130
implicit, 21
TransitionState, 109
transposition
matrix and vector, 7
TSX 60, 131
TSX Composite, 172
daily model, 204
monthly model, 202
Tukey's honestly significant differences, 196
turnover, 20, 37

United States, 172
fundamental variables, 199
universal portfolios, 64
UseOnTrain, 108
utility function, 9, 15
non-standard, 56
utility maximization, 14

validation set, 76
value function, 78
approximation, 80
multiperiod portfolio choice, 39
parametric approximation, 85
tabular representation, 83
value iteration, 79
value-at-risk, 60
risk measure, 17
Vapnik–Chervonenkis dimension, 75
VAR, *see* vector autoregressive model
VaR, *see* value-at-risk
variance, 2, 74
vector
all ones, 7
vector autoregressive model
expected stock returns, 28
views
of market by investor in Black–Litterman model, 34
vintage data, 97
database management, 97
learning algorithms, 97
volatility filter
technical trading rule, 115

wealth dynamics
in multiperiod portfolio choice, 39
Wiener process, 298
worst month
performance measure, 179