The K-Shortest-Paths Approach to Approximate Dynamic Programming

Nicolas Chapados and Yoshua Bengio
{chapados,bengioy}@iro.umontreal.ca
Université de Montréal
C.P. 6128, Montral, Qubec, H3C 3J7

Markov Decision Problems

- Discrete-time system evolving according to

$$x_{t+1} = f_t(x_t, u_t, w_t)$$

- Current state: x_t Action: u_t Disturbance: w_t
- At each step, incur a **cost** $g_t(x_t, u_t, w_t)$
- **OBJECTIVE**: minimize the total expected cost (over, in this setting, the finite horizon $t = 1, \ldots, T$):

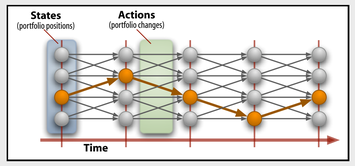
$$J_0^*(x_0) = \min_{u_1,\ldots,u_{T-1}} \mathop{\mathrm{E}}_{w_1,\ldots,w_{T-1}} \left[g_T(x_T) + \sum_{t=1}^{T-1} g_t(x_t, u_t, w_t) \middle| x_0 \right]$$

Dynamic Programming

- The finite-horizon problem solved with Bellman's recursion:

$$\begin{aligned} J_T^*(x_T) &= g_T(x_T) \\ J_t^*(x_t) &= \min_{u_t} \mathop{\mathrm{E}}_{w_t} \left[g_t(x_t, u_t, w_t) + J_{t+1}^*(f_t(x_t, u_t, w_t)) \right] \end{aligned}$$

- Equivalent to shortest-cost path in the state-action graph:



Portfolio Optimization

- Goal: manage a portfolio (e.g. stocks, futures)
- We can invest in a universe of M assets
- State: $x_t = (n_{t,1}, \ldots, n_{t,M}, p_{t,1}, \ldots, p_{t,M})$, where
  - $n_{t,i} \in \mathbf{Z}$: number of shares of asset i held
  - $p_{t,i} \in \mathbf{R}_+$: price of asset i at time t
- Possible actions: $u_t \in \mathbf{Z}^M$, i.e. buy or sell an integral number of shares (short positions allowed)
- The cost $g_t(x_t, u_t)$ at time t is the $ amount required to carry out u_t (i.e. establish the desired position), accounting for transaction costs and market slippage
- This cost function yields the **highest-profit** portfolio

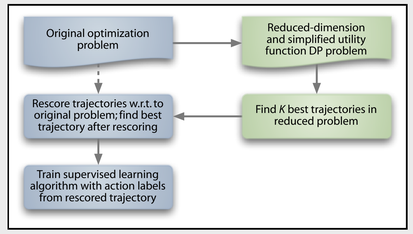
Problem: Additive Utilities

- In portfolio allocation, we wish to **maximize expected utility**
- Dynamic programming works for additive utility functions, where the total utility can be expressed as a sum of individual (timewise) utilities
- However, many useful risk-averse utility functions in finance must take the **entire trajectory** into account, for example the Sharpe Ratio (average excess return divided by standard deviation of returns), or statistics that consider the **maximum drawdown** (relative loss between the maximum and minimum portfolio values at any point during the investment horizon)

Proposed Solution: Rescoring

- We assume that given a trajectory, we can efficiently compute any utility function of interest
- **Idea in a nutshell:** we extract a large number of "good" trajectories under the *wrong* utility function, and we **rescore** (re-rank) them under the *desired* utility. For convenience, call the first one the *source utility*, and second one is called the *target utility*
- In portfolio management, the source utility is the maximal profit; the target utility is generally a risk-adjusted measure such as the Sharpe Ratio or the Sortino Ratio
- Significantly, the target utility **needs not be additive**; any path-dependent utility function can be used
- This idea is used successfully in speech recognition [2]

The Controller Big Picture



Enumerating the k-Shortest Paths

- We rely on the efficient *Recursive Enumeration Algorithm* recently proposed by Jimnez and Marzal [1]
- Let $\pi^k(v)$ denote k-th best path ending at vertex v and starting at constant vertex s, and let $L^k(v)$ denote its length; similarly, we denote by $L(\pi)$ the length of path π
- Let $\Gamma^{-1}(v)$ denote the predecessors of v in the graph; moreover, denote by $\pi \cdot u$ the operation of *path extension*, i.e. path π extended to vertex u
- As explained in [1], this algorithm relies on a generalization of Bellman's recursion, wherein, for all $v \in$ the graph V,

$$L^k(v) = \begin{cases} 0, & \text{if } k = 1 \text{ and } v = s; \\ \min_{\pi \in C^k(v)} L(\pi), & \text{otherwise;} \end{cases}$$

$$\pi^k(v) = \begin{cases} s, & \text{if } k = 1 \text{ and } v = s; \\ \arg\min_{\pi \in C^k(v)} L(\pi), & \text{otherwise;} \end{cases}$$

where if $k = 1$ and $v \neq s$, or $k = 2$ and $v = s$, then

$$C^k(v) = \{\pi^1(u) \cdot v : u \in \Gamma^{-1}(v)\};$$

otherwise, if u and k' are the vertex and index, respectively, such that $\pi^{k-1}(v) = \pi^{k'}(u) \cdot v$, then

$$C^k(v) = \left(C^{k-1}(v) - \{\pi^{k'}(u) \cdot v\} \right) \cup \{\pi^{k'+1}(u) \cdot v\}$$

Rescoring Example

- Four assets (futures on BritishPound, Sugar, Silver, HeatingOil); allow from -3 to $+3$ shares of each asset in portfolio; maximum variation of $+1$ or -1 share per time-step; proportional transaction costs of 0.5%
- Extract the best 2.5×10^6 paths under the "Terminal Wealth" utility, rescore under the "Average Return" and "Sharpe Ratio" utilities; 30-time-step price history

figure not recovered
(`asset_prices`)

figure not recovered
(`positions_max_e`)

figure not recovered
(`positions_max_sharpe`)

More on k-Shortest Paths

- *Intuition*: keep all partial results from first-best-path algorithm (e.g. Dijkstra). Second best path to x_t is obtained from the best of: (i) best path to non-best predecessor $\{C_1, C_2, \ldots, \}$, (ii) Second best path to best predecessor Y (recursively).
- *Efficiently enumerating the actions*: from a vertex x_t, the action (position delta vector) yielding the **best absolute return** to x_{t+1} is that having the largest dot product with the asset return vector from t to $t + 1$. Hence, by enumerating the actions in **order of decreasing dot product** with the return vector r_{t+1}, we can achieve significant pruning in the action enumeration.

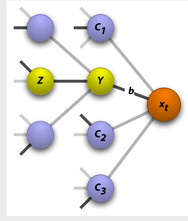
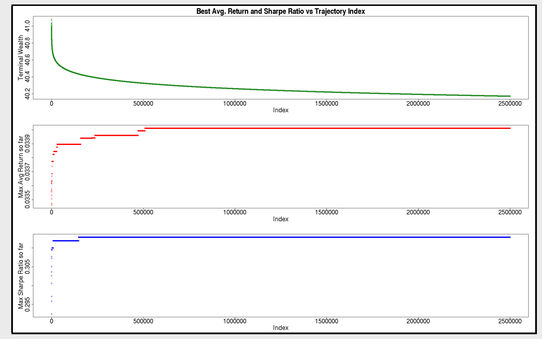


figure not recovered
(`state_action`)

Utility Versus Path Index

Correlation structure of utility functions for the sample price history:

	Terminal Wealth	Avg. Return	Sharpe Ratio
Terminal Wealth	1.00	0.36	0.13
Avg. Return	0.36	1.00	0.45
Sharpe Ratio	0.13	0.45	1.00

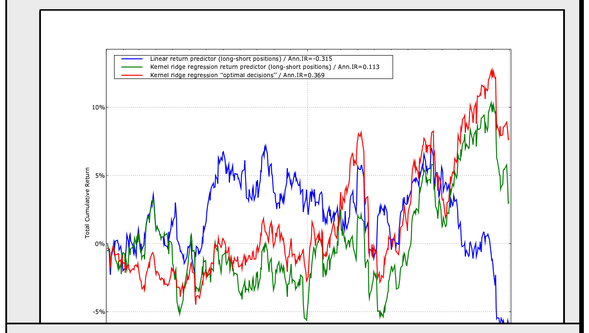


Asset Allocation Experiment

- Four assets (futures on FeederCattle, Cotton, Corn, Silver); allow from -1 to $+1$ shares of each asset in portfolio; proportional transaction costs of 0.5%
- Compare (i) *return forecasting models* (target is the 10-day ahead 22-day asset returns) against (ii) *optimal decision under Sharpe Ratio utility function* (target is $-1/0/+1$ as determined from rescoring)
- Forecasting model is *Kernel Ridge Regression*:

$$f(\mathbf{x}) = k(\mathbf{x}, \mathbf{x}_i)(M + \lambda I)^{-1}\mathbf{y}$$

where $k(\mathbf{x}, \mathbf{x}_i)$ is the vector of kernel evaluations of the test vector $\mathbf{x}$ against all elements of the training set $\mathbf{x}_i$, M is the Gram matrix on the training set, and $\mathbf{y}$ is the matrix of targets. Used a Gaussian kernel with $\sigma = 3.0$ and fixed $\lambda = 10.0$.



[1] V. Jimnez and A. Marzal, *Computing the K Shortest Paths: a New Algorithm and an Experimental Comparison*, In "Algorithm Engineering", J. S. Vitter and C. D. Zaroliagis (eds.), Springer-Verlag, vol. 1668, pp. 15–29, 1999

[2] L. Rabiner and B. H. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.