Extending Metric-Based Model Selection and Regularization in the Absence of Unlabeled Data

DIRO Technical Report #1200

Yoshua Bengio and Nicolas Chapados
Dept. IRO, Université de Montréal
C.P. 6128, Montreal, Qc, H3C 3J7, Canada
{*bengioy,chapados*}*@iro.umontreal.ca*

3 July 2001

Abstract

Metric-based methods have recently been introduced for model selection and regularization, often yielding very significant improvements over all the alternatives tried (including cross-validation). However, these methods require a large set of unlabeled data, which is not always available in many applications. In this paper we extend these methods (TRI, ADJ and ADA) to the case where no unlabeled data is available. The extended methods (xTRI, xADJ, xADA) use a model of the input density directly estimated from the training set. The intuition is that the main reason why the above methods work well is that they make sure that the learned function behaves similarly on the training points and on "neighboring" points. The experiments are based on estimating a simple non-parametric density model. They show that the extended methods perform comparably to the originals even though no unlabeled data is used.

1 Model Selection and Regularization

Supervised learning algorithms take a finite set of input/output training pairs $\{(x_1, y_1) \cdots (x_l, y_l)\}$ sampled (usually independently) from an unknown joint distribution $P(X, Y)$ and attempt to infer a function $f \in \mathcal{F}$ that minimizes the expectation of the loss $L(f(X), Y)$ (also called the *generalization error*. In many cases one faces the dilemma that if $\mathcal{F}$ is too "rich" then the average training set loss (*training error*) will be low but the expected out-of-sample loss may be large (*overfitting*), and vice-versa if $\mathcal{F}$ is not "rich" enough (*underfitting*).

In many cases one can define a collection of increasingly complex function classes $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \cdots \subset \mathcal{F}$. *Model selection* methods attempt to choose one

of these function classes to avoid both overfitting and underfitting. For example with complexity penalization methods (Rissanen, 1986; Foster and George, 1994), one first applies the learning algorithm to each of the function classes (on the whole training set), yielding a sequence of hypothesis functions $f_0, f_1, \ldots$. Then one of them is chosen based on a criterion that combines average training loss and a measure of complexity (which usually depends only on the function class $\mathcal{F}_i$). Another type of model selection is based on *held-out data*: one estimates the generalization error by repeatedly training on a subset of the data and testing on the rest (e.g. using the bootstrap, leave-one-out or K-fold cross-validation). One uses this generalization error estimator in order to choose the function class that appears to yield the lowest generalization error. Cross-validation estimators have been found to be very robust and difficult to beat (they estimate generalization error almost unbiasedly, but possibly with large variance).

With *regularization* methods one looks for a function in the larger class $\mathcal{F}$ that minimizes a training criterion that combines both the average training loss and a penalization for complexity (e.g. minimizing curvature (Poggio and Girosi, 1990)). These approaches are not restricted to a discrete set of complexity classes and may potentially strike a better compromise between fitting the training data and avoiding overfitting.

An overview of advances in model selection and model combination methods can be found in a recent Machine Learning special issue (Bengio and Schuurmans, 2001).

2 Metric-Based Model Selection and Regularization

Metric-based methods for model selection and regularization are based on the idea that solutions that overfit are likely to behave very differently on the training points and on other points sampled from the input density $P_X(x)$. This occurs because the learning algorithm tries to reduce the loss at the training points (but not necessarily elsewhere since no data is available there), whereas we want the solution to work well not only on the training points but in general where $P_X(x)$ is not small. These methods are all based on the definition of a *metric* (or pseudo-metric) on the space of functions, which allows to judge how far two functions are from each other:

$$d(f, g) = \psi(E[L(f(X), g(X))])$$

where the expectation $E[\cdot]$ is over $P_X(x)$ and ψ is a normalization function. For example with the quadratic loss $L(u, v) = (u - v)^2$, the proper normalization function is $\psi(z) = z^{1/2}$. Although $P_X(x)$ is unknown, Schuurmans (1997) proposed to estimate $d(f, g)$ using an average computed on *unlabeled data* (i.e. points x_i sampled from $P_X(x)$ but for which no associated y_i is given). In what follows we shall use $d(f, g)$ to denote the distance estimated on a (large) unlabeled set. The metric-based methods are based on comparing $d(f, g)$ with the

corresponding average distance measured *on the training set*:

$$\hat{d}(f,g) = \psi(\frac{1}{l}\sum_{i=1}^{l} L(f(x_i), g(x_i)))$$

In addition, the following notation is introduced to compare a function to the "truth":

$$d(f, P_{Y|X}) = \psi(E[L(f(X), Y)]$$

and on the training set

$$\hat{d}(f, P_{Y|X}) = \psi(\frac{1}{l}\sum_{i=1}^{l} L(f(x_i), y_i)).$$

Schuurmans (1997) first introduced the idea of a metric-based model selection by taking advantage of possible violations of the *triangle inequality*, with the **TRI model selection algorithm**, which chooses the last hypothesis function f_l (in a sequence $f_0, f_1, \ldots$ of increasingly complex functions) that satisfies the *tri*angle inequality $d(f_k, f_l) \leq \hat{d}(f_k, P_{Y|X}) + \hat{d}(f_l, P_{Y|X})$ with every preceding hypothesis f_k, $0 \leq k < l$. Improved results were obtained with a new penalization model selection method, based on similar ideas, called **ADJ**, which chooses the hypothesis function f_l which minimizes the *adj*usted loss

$$\hat{\hat{d}}(f_l, P_{Y|X}) = \hat{d}(f_l, P_{Y|X}) \max_{k<l} \frac{d(f_k, f_l)}{\hat{d}(f_k, f_l)},$$

again using unlabeled data to compute $d(f_k, f_l)$. See (Schuurmans and Southey, 2001) for more detailed justification and for experiments showing that these two methods outperform classical model selection procedures (including cross-validation) on many small artificial data sets (with between 10 and 30 training examples) on which overfitting can be severe.

More recently, this idea of a metric-based method to control overfitting was extended to the regularization paradigm with the **ADA** algorithm, which in the case of regression is defined by the following regularized training criterion:

$$\min_{f \in \mathcal{F}} \hat{d}(f, P_{Y|X}) \max(\frac{d(f,\phi)}{\hat{d}(f,\phi)}, \frac{\hat{d}(f,\phi)}{d(f,\phi)})$$

where ϕ is a kind of prior function which can be chosen for example as the constant average of the y_i's. Even better results were obtained with this method, and comparisons were performed (Schuurmans and Southey, 2001) that show ADA not only to beat in most cases a wide array of model selection and regularization techniques, but also to often beat the oracle model selection method (that picks f_i that minimizes $d(f_i, P_{Y|X})$ out of a finite set). This may occur because ADA is a regularization technique that can thus make a finer trade-off between overfitting and underfitting.

3 Proposed Extension when No Unlabeled Data is Available

In this paper we propose an extension to TRI, ADJ and ADA in the case where no unlabeled data is available, since this occurs in many applications. The basic idea is to approximate the expected value of $d(f, g)$ using a model of the input density rather than with an average over unlabeled data:

$$\tilde{d}(f, g) = \psi(\int \hat{P}_X(x) L(f(x), g(x)) dx)$$

where $\hat{P}_X(x)$ is the model of the input density. This model may be derived from the training set input points $(x_1, \ldots, x_l)$ and/or prior knowledge about the input density. In our experiments we have simply used a Parzen windows density estimator (that only relies on the data):

$$\hat{P}_X(x) = \frac{1}{l} \sum_i \frac{e^{-0.5||x_i - x||^2/\sigma^2}}{(2\pi)^{n/2}\sigma^n}$$

where $x \in \mathcal{R}^n$. Such a model leaves the smoothing parameter to be chosen. Many methods have been proposed for setting the smoothing parameter of such non-parametric density estimators. In our experiments we have experimented with the effect of varying σ and we have used an asymptotically motivated estimator (Scott, 1992):

$$\sigma = \frac{1.144\sqrt{\hat{Var}[X]}}{l^{1/5}}. \quad (1)$$

where $\hat{Var}[X]$ denotes sample variance of the inputs on the training set.

3.1 Extension of Theoretical Results

In (Schuurmans and Southey, 2001), several attractive theoretical results have been proved concerning the generalization performance of TRI, ADJ and ADA. Almost identical results can be proved for the extended algorithms that use an estimated input density $\hat{P}_X(x)$, with the following change: instead of characterizing the true generalization error (i.e. averaging the error over the true input density $P_X(x)$), those results characterize the generalization error measured with respect to the estimated input density $\hat{P}_X(x)$. The main theoretical results are briefly recalled here. Let us call xTRI, xADJ, and xADA the extended methods using $\hat{P}_X(x)$.

Proposition 1 Let $f_m = \text{argmin}_{f_i} \tilde{d}(f_i, P_{Y|X})$, and let f_l be the hypothesis selected by xTRI. If $m \leq l$ and $\hat{d}(f_m, P_{Y|X}) \leq \tilde{d}(f_m, P_{Y|X})$ then $\tilde{d}(f_l, P_{Y|X}) \leq 3\tilde{d}(f_m, P_{Y|X})$.

The above tells us when xTRI cannot overfit too much.

Proposition 2 Let $f_m = \text{argmin}_{f_i} \tilde{d}(f_i, P_{Y|X})$, and let f_l be the hypothesis selected by xADJ. If $m \leq l$ and $\hat{\tilde{d}}(f_m, P_{Y|X}) \leq \tilde{d}(f_m, P_{Y|X}))$ then $\tilde{d}(f_l, P_{Y|X}) \leq (2 + \frac{\hat{d}(f_m, P_{Y|X})}{\hat{d}(f_l, P_{Y|X})})\tilde{d}(f_m, P_{Y|X})$.

The above tells us when xADJ cannot overfit too much.

Proposition 3 Consider a hypothesis f_m, and assume that $\hat{d}(f_l, P_{Y|X}) \leq \tilde{d}(f_l, P_{Y|X})$ for $l \leq m$, and $\hat{d}(f_l, P_{Y|X}) \leq \hat{\tilde{d}}(f_l, P_{Y|X})$ for $l \leq m$. Then if $\tilde{d}(f_m, P_{Y|X}) \leq \frac{1}{3}\frac{\hat{d}(f_l, P_{Y|X})^2}{d(f_l, P_{Y|X})}$ for $l < m$ (i.e. $\tilde{d}(f_m, P_{Y|X})$ is sufficiently small) it follows that $\hat{\tilde{d}}(f_m, P_{Y|X}) < \hat{\tilde{d}}(f_l, P_{Y|X})$ for $l < m$ and therefore xADJ will not choose any predecessor in lieu of f_m.

The above tells us when xADJ cannot underfit too much. See (Schuurmans and Southey, 2001) for proofs of equivalent results for the original methods, and detailed discussions. The same authors observe that the conditions required for these propositions to hold are not always satisfied in practice.

4 Experimental Results

4.1 Artificial Data Results

To ascertain the validity of the proposed extensions, we first performed experiments on artificial data, keeping the same overall framework as (Schuurmans and Southey, 2001). We tested the methods on data generated from the four following functions:

$f_1(x) = \mathbf{1}_{x \geq 0.5}$ $\quad$ $f_3(x) = \sin^2(2\pi x)$

$f_2(x) = \sin(1/x)$ $\quad$ $f_4(x) = 32x - 275x^2 + 777x^3 - 892x^4 + 358x^5$

Our class of nested models is the set of polynomials of fixed order,

$$f(x; \beta) = \sum_{j=0}^{m} \beta_j x^j,$$

where the order m depends on the size of the training set (described below). The models are trained to minimize a squared error criterion with weight decay; the members of the class vary according to the value of the weight-decay coefficient λ (a higher λ implies a reduced capacity). We restricted λ to the set $\{10^i : i = -5, -4.5, -4, \ldots, 2\}$, which was found to be adequate. The optimal parameter vector $\hat{\beta}^*$ is given by the well-known *ridge estimator*, $\hat{\beta}^* = (X'X + \lambda I)^{-1} X'Y$, where X is the matrix of regressors, Y is the (column-) vector of targets, and I is the identity matrix.

The artificial data sets are generated from the functions $f_1, \ldots, f_4$. The input density $P_X(x)$ is chosen as either uniform on the $[0, 1]$ interval ($U(0, 1)$), or normal with mean $\frac{1}{2}$ and variance $\frac{1}{12}$ ($N(\frac{1}{2}, \frac{1}{12})$, chosen to have the same mean and variance as the $U(0, 1)$ distribution). Additive noise with a distribution $N(0, 0.0025)$ is added in every case.

As in (Schuurmans and Southey, 2001), training sets of sizes $\{10, 20, 30\}$ are considered. The size of the unlabeled sets generated from $P_X(x)$ (for TRI, ADJ, and ADA) is fixed to 200. The same number of unlabeled examples are sampled from the Parzen estimator for xTRI, xADJ, and xADA. Finally, a single test set of size 5000 is generated to evaluate performance.

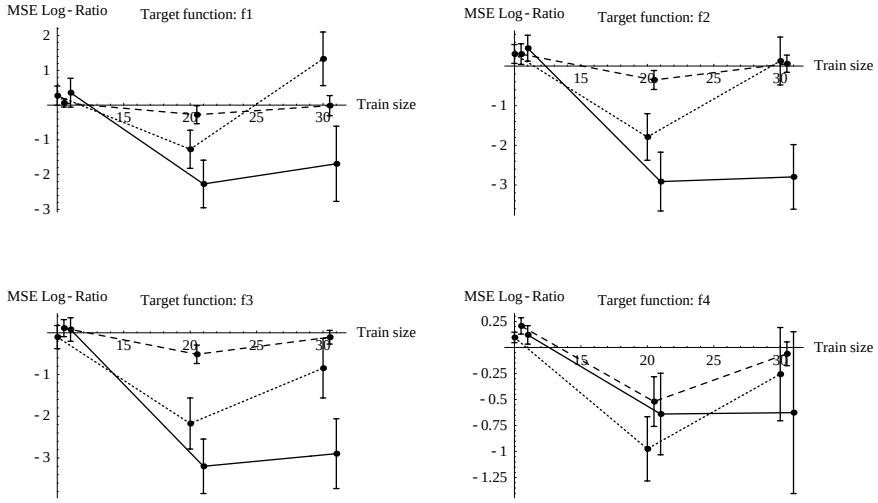

Figure 1: *Comparison between Parzen-generated unlabeled data and "true" unlabeled data, for xTRI/TRI (dotted line), xADJ/ADJ (dashed line), and xADA/ADA (solid line). Each point is the median test MSE log-ratio of 50 repetitions, with random training, unlabeled and test sets; the error bars represent 1 standard error on the median, computed by bootstrap resampling. The input distribution is* $N(\frac{1}{2}, \frac{1}{12})$*. Surprisingly, the Parzen-generated data sometimes performs* ***significantly better*** *(negative log-ratio) than unlabeled data from "true"* $P_X(x)$*, and never significantly worse.*

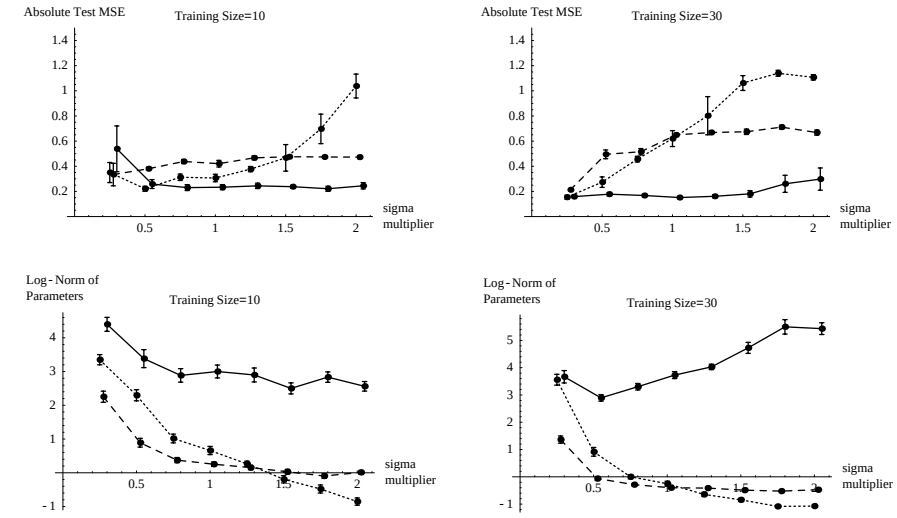

Figure 2: *Effect of Parzen estimator window width for xTRI (dotted line), xADJ (dashed line), and xADA (solid line).* ***Panel (a)*** *shows the median MSE performance; we observe that xTRI is very sensitive to the window width, xADJ less to, and that xADA is nearly insensitive to it.* ***Panel (b)*** *shows the log-norm of the parameters of the trained models; we note that the window width has a clear regularization effect for xTRI and xADJ.* ***In both cases****, the window width is given as a multiplicative factor applied to eq.* (1)*. All results are computed for function f1 under a* $U(0,1)$ *input distribution, 50 repetitions. The error bars represent 1 standard error on the median and the mean respectively.*

Table 1: *Mean log ratio of each method MSE to the 10-fold cross-validation MSE (standard errors are in parentheses). Negative results indicate better performance than cross-validation. Proposed algorithms perform comparably to the originals. 50 repetitions of each experiment are performed, under the normal input distribution. l is the training set size.*

	l	TRI	xTRI	ADJ	xADJ	ADA	xADA
	10	-2.06 (0.43)	-1.79 (0.36)	-2.27 (0.36)	-2.20 (0.37)	-1.65 (0.43)	-1.30 (0.41)
f_1	20	-4.28 (0.73)	-5.55 (0.51)	-5.57 (0.58)	-5.85 (0.53)	-1.96 (0.69)	-4.23 (0.62)
	30	-7.72 (1.13)	-6.39 (0.99)	-7.94 (1.04)	-7.96 (1.04)	-1.36 (1.14)	-3.05 (1.33)
	10	-1.34 (0.33)	-1.03 (0.28)	-1.63 (0.33)	-1.33 (0.28)	-1.11 (0.31)	-0.66 (0.31)
f_2	20	-3.02 (0.81)	-4.81 (0.46)	-5.15 (0.59)	-5.51 (0.54)	-0.12 (0.75)	-3.03 (0.64)
	30	-5.73 (0.77)	-5.61 (0.69)	-6.69 (0.77)	-6.63 (0.74)	1.51 (0.99)	-1.28 (0.93)
	10	-0.73 (0.46)	-0.84 (0.39)	-1.98 (0.43)	-1.87 (0.37)	-1.61 (0.41)	-1.53 (0.39)
f_3	20	-2.87 (0.96)	-5.05 (0.71)	-5.60 (0.80)	-6.12 (0.71)	-0.98 (0.96)	-4.18 (0.88)
	30	-6.36 (1.11)	-7.21 (0.91)	-7.36 (0.93)	-7.47 (0.89)	0.14 (1.20)	-2.75 (1.14)
	10	-0.53 (0.13)	-0.44 (0.13)	-0.52 (0.14)	-0.31 (0.12)	-0.40 (0.18)	-0.28 (0.15)
f_4	20	-1.99 (0.48)	-2.96 (0.47)	-2.39 (0.53)	-2.91 (0.50)	-2.13 (0.52)	-2.77 (0.47)
	30	-4.04 (0.83)	-4.29 (0.76)	-4.19 (0.79)	-4.25 (0.81)	-1.26 (0.88)	-1.89 (0.97)

Table 4.1 gives detailed results comparing each method against 10-fold cross-validation. We observe that the proposed methods give comparable results to the original ones. Similar results are borne out of figure 4, which directly compares the proposed methods (xTRI, xADJ, and xADA) against the originals (TRI, ADJ, and ADA).

Figure 4 shows the effect of the Parzen estimator bandwidth σ. Panel (a) shows the sensitivity of the median MSE of the three algorithms to σ; xADA is the least sensitive to the choice of σ. Panel (b) shows the sensitivity of the log-norm of the parameters (i.e. reflecting the complexity of the solution) with respect to σ. In most cases, as we expected, a higher σ yields simpler solutions. However, unexpectedly, we find in one case that larger σ yields larger parameters (for xADA).

4.2 Results on the ABALONE Dataset

We also performed a series of experiments on a non-artificial regression task (ABALONE) from the UCI Machine Learning Repository (using 1000 examples). We used for this purpose a RBF network regularized with weight decay, with one RBF per training example:

$$f(z;\sigma) = \sum_{i=1}^{l} w_i \exp\left(-\frac{\|z - x_i\|^2}{2\sigma^2}\right), \tag{2}$$

where $\{x_i\}_{i=1}^{l}$ are the training set inputs, and σ is the RBF window width.[1] For a given weight decay λ, an analytical solution for the w_i that minimizes

[1] Not to be confused with the window width of the Parzen distribution used to sample unlabeled examples!

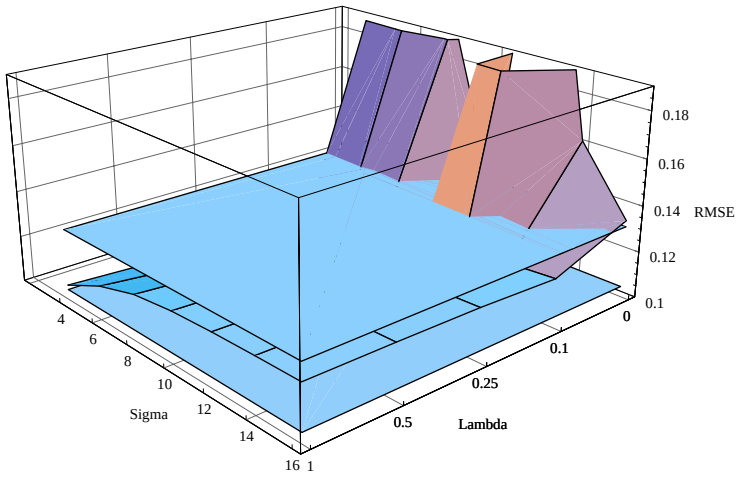

Figure 3: *Performance comparison betwen ADA, xADA, and fixed regularizers for the* ABALONE *regression task, using a RBF network. The irregular surface on the graph plots the RMSE at various RBF window widths (sigma) and weight decays (lambda). The upper horizontal cutting plane gives the mean RMSE for ADA (0.131). The lower horizontal cutting plane gives the mean RMSE for xADA (0.104). All results are the average of 100 repetitions, each time resampling randomly new disjoint training sets (10% of total data), unlabeled (70%, for ADA only) and test sets (20%). Not shown on the graph, xADA performs nearly always statistically significantly better than any fixed regularizer, and never significantly worse; xADA is also significantly better than ADA.*

the squared error criterion is easily obtained (similar to the ridge estimator presented in section 4.1).

Figure 4.1 gives more details and shows that xADA performs surprisingly well, outperforming both ADA and nearly every fixed regularizer.

5 Conclusion

We have proposed extensions of the metric-based model selection (TRI, ADJ) and regularization (ADA) methods that does not require unlabeled data. These extensions (xTRI, xADJ, and xADA) are based on an estimated model of the input distribution; our experiments show that even a simple non-parametric density model yields performance comparable to the original methods (even though the latter have access to extra information in the form of unlabeled data).

Some questions remain unanswered: Why do the proposed methods actually perform better in some cases? How does the complexity of solution behave in terms of the smoothness of the estimated density, especially in the case of xADA? Future work should address these questions and confirm the present results on a wider range of data sets.

References

Bengio, Y. and Schuurmans, D. (2001). Special issue on new methods for model selection and model combination.

Foster, D. and George, E. (1994). The risk inflation criterion for multiple regression. *Annals of Statistics*, 22:1947–1975.

Poggio, T. and Girosi, F. (1990). Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982.

Rissanen, J. (1986). Stochastic complexity and modeling. *Annals of Statistics*, 14:1080–1100.

Schuurmans, D. and Southey, F. (2001). Metric-based methods for adaptive model selection and regularization.

Scott, D. W. (1992). *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, New York.