Appears in
Proceedings of the Seventeenth International Conference on Machine Learning (ICML-2000),
Stanford, CA, June 2000.

An Adaptive Regularization Criterion for Supervised Learning

Dale Schuurmans DALE@CS.UWATERLOO.CA
Finnegan Southey FDJSOUTHEY@CS.UWATERLOO.CA
Department of Computer Science, University of Waterloo, Waterloo, Ontario N2L 3G1, Canada

Abstract

We introduce a new regularization criterion that exploits *unlabeled* data to adaptively control hypothesis-complexity in general supervised learning tasks. The technique is based on an abstract metric-space view of supervised learning that has been successfully applied to model selection in previous research. The new regularization criterion we introduce involves no free parameters and yet performs well on a variety of regression and conditional density estimation tasks. The only proviso is that sufficient unlabeled training data be available. We demonstrate the effectiveness of our approach on learning radial basis functions and polynomials for regression, and learning logistic regression models for conditional density estimation.

1. Introduction

In the canonical supervised learning task one is given a training set $\langle x_1, y_1\rangle, ..., \langle x_t, y_t\rangle$ and attempts to infer a hypothesis function $h : X \to Y$ that achieves a small prediction error $err(h(x), y)$ on future test examples. This general paradigm covers many of the problems studied in machine learning research, including: *regression*, where Y is typically $\mathbb{R}$ and we measure prediction error by squared difference or some similar loss $err(\hat{y}, y) = (\hat{y} - y)^2$; *classification*, where Y is typically a small discrete set and we measure prediction error with the misclassification loss $err(\hat{y}, y) = 1_{(\hat{y} \neq y)}$; and *conditional density estimation*, where we assume, for example, that Y is a classification label in $\{0, 1\}$ and $\hat{Y}$ is a probabilistic prediction in $[0, 1]$, and we measure prediction error using the log loss $err(\hat{y}, y) = -y \log \hat{y} - (1 - y) \log(1 - \hat{y})$.

Regardless of the specifics of these scenarios, one always faces the classic over-fitting versus under-fitting dilemma in supervised learning: If the hypothesis is chosen from a class that is too complex, there is a good chance the hypothesis will demonstrate large test error even when its training error is small. This is because complex classes generally contain several hypotheses that behave similarly on the training data and yet behave quite differently in other parts of the domain—which destroys the ability to distinguish good hypotheses from bad. (Note that significantly different hypotheses cannot all be simultaneously accurate.) Therefore, one must restrict the hypothesis class in order to reliably distinguish good from bad hypotheses. On the other hand, choosing hypotheses from an overly restricted class might prevent one from being able to express a good approximation to the ideal predictor and thereby cause important structure in the training data to be ignored. Since both under-fitting and over-fitting result in large test error, they must be avoided simultaneously.

This of course is an old issue, and a variety of techniques have been proposed for coping with the fundamental tradeoff. In this paper, we are primarily interested in investigating *automated* methods for calibrating hypothesis complexity for given training data. Most of the techniques that have been developed for this problem fall into one of three basic categories: model selection, regularization, and model averaging.

In *model selection* one first takes a base hypothesis class, H, decomposes it into a discrete collection of subclasses $H_0 \subset H_1 \subset \cdots = H$, and then, given training data, attempts to identify the optimal subclass from which to choose the final hypothesis. There have been a variety of methods proposed for choosing the optimal subclass, but most techniques fall into one of two basic categories: *complexity penalization* (e.g., the minimum description length principle (Rissanen, 1986) and various statistical selection criteria (Foster & George, 1994)); and *hold-out testing* (e.g., cross-validation and bootstrapping (Efron, 1979)).

Regularization is similar to model selection except that one does not impose a discrete decomposition on the base hypothesis class. Instead a penalty criterion is imposed on the individual hypotheses, which either penalizes their parametric form (e.g., as in ridge regression or weight decay in neural network training (Cherkassky & Mulier, 1998; Ripley, 1996; Bishop, 1995)) or penalizes their global smoothness properties (e.g., minimizing curvature (Poggio & Girosi, 1990)).

Model averaging methods do not select a single hypothesis but rather take a weighted combination of base hypotheses to form a composite predictor. Composing base functions in this way can have the effect of smoothing out erratic hypotheses (e.g., as in Bayesian model averaging (MacKay, 1992) and bagging (Breiman, 1996)), or increasing the representation power of the base hypothesis class via linear combinations (e.g., as in boosting (Freund & Schapire, 1997) and neural network ensemble methods (Krogh & Vedelsby, 1995)).

All of these methods have shown impressive improvements over naive learning algorithms in every area of supervised learning research. However, one difficulty with these techniques is that they usually require expertise to apply properly, and often involve free parameters that must be set by an informed practitioner.

In this paper we introduce a new regularization criterion that *automatically* chooses the right complexity of hypothesis to fit to given training data. The idea is to use *unlabeled* data to penalize hypotheses that behave erratically off the training set. We show how unlabeled data can be used to tune the degree of regularization for a given task without having to set free parameters by hand. In particular, we show that this technique automatically adjusts its behavior to a given training set and can thereby outperform fixed regularizers for a given task. The technique we propose extends earlier work on model selection (Schuurmans, 1997; Schuurmans et al., 1997) but exhibits superior performance and can be applied in a wider range of contexts.

2. Geometry of Supervised Learning

Throughout the paper we adopt a metric space view of supervised learning that was introduced in (Schuurmans, 1997). Assume that the examples $\langle x, y\rangle$ are generated by a stationary joint distribution P_{XY} on $X \times Y$. In learning a hypothesis function $h : X \to Y$ we are primarily interested in modeling the conditional distribution $P_{Y|X}$. However, here we explore the utility of using extra information about the marginal domain distribution P_X to choose a better hypothesis. Note that information about P_X can be obtained from a collection of *un*labeled training examples $x_1, ..., x_r$. (These are often in abundant supply in many applications—e.g., text processing and computer perception.) The significance of having information about the domain distribution P_X is that it defines a natural *metric structure* over the space of hypotheses. That is, for any two hypothesis functions f and g we can obtain a natural measure of the distance between them by computing the expected disagreement in their predictions

$$d(f,g) \;\triangleq\; \text{“}\int err(f(x), g(x))\, dP_X\text{”} \qquad (1)$$

For example, in regression we measure the distance between two prediction functions by[1]

$$d(f,g) \;=\; \left(\int (f(x)-g(x))^2\, dP_X\right)^{1/2} \qquad (2)$$

In classification, we measure the distance between two classifiers by

$$d(f,g) \;=\; \int 1_{(f(x)\neq g(x))}\, dP_X \;=\; P_X(f(x)\neq g(x))$$

In conditional density estimation one can measure the "distance" between two conditional probability models (that map to predictions in $[0,1]$) by their Kullback-Leibler divergence[2]

$$d(f\|g) = \int -f(x)\log\frac{f(x)}{g(x)} - (1-f(x))\log\frac{1-f(x)}{1-g(x)}\; dP_X$$

In each case, the resulting distances can be efficiently calculated by making a single pass down a list of unlabeled examples.

Note that the generic definition of distance given in (1) can be further extended to include the target conditional distribution in an analogous manner

$$d(f, P_{Y|X}) \;\triangleq\; \text{“}\iint err(f(x), y)\, dP_{Y|x}\, dP_X\text{”} \qquad (3)$$

So, in regression this yields the root mean squared error of a hypothesis, in classification it gives the true misclassification probability, and in conditional probability modeling it gives the expected log loss (or KL-divergence to $P_{Y|X}$). Thus, definitions (1) and (3) show how, using unlabeled data, one can generically embed the entire supervised learning problem into a metric space structure: given labeled training examples $\langle x_1, y_1\rangle, ..., \langle x_t, y_t\rangle$, the goal is to find the hypothesis $h \in H$ that is closest to a target conditional $P_{Y|X}$ while

[1] Note that the square root is necessary to achieve a metric in this case (hence the need for the scare quotes above). However, many natural loss functions can be renormalized to recover the triangle inequality in a similar way.

[2] Although technically this is not a metric, since KL-divergence is neither symmetric nor satisfies the triangle inequality, it nevertheless supplies a useful measure.

using only estimates of the distance $d(h, \mathrm{P}_{Y|X})$ given by the estimated distances

$$\hat{d}(h, \mathrm{P}_{Y|X}) \;\triangleq\; \text{``}\sum_{i=1}^{t} err(h(x_i), y_i)/t\text{''}$$

Earlier work (Schuurmans, 1997) showed how these metric notions could be used to devise effective model selection procedures. There, the basic idea was to use the triangle inequality to detect when a series of empirical error estimates, $\hat{d}(h_0, \mathrm{P}_{Y|X}), \hat{d}(h_1, \mathrm{P}_{Y|X}), \ldots$, were no longer trustworthy, and stop the selection process at an appropriate hypothesis h_i in the model selection sequence. Here we extend these ideas to a more general regularization criterion that uses the unlabeled data to decide how to penalize *individual* hypotheses.

3. An Adaptive Regularization Criterion

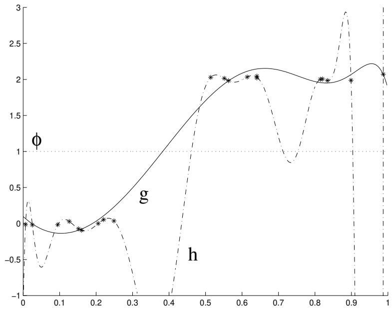
The main contribution of this paper is a simple, generic training objective that can be applied to a wide range of supervised learning problems. Continuing from above, we assume that we have access to a sizable collection of unlabeled data, which we now use to penalize complex hypotheses. The intuition behind our criterion is simple: instead of minimizing empirical training error alone, we in addition seek hypotheses that behave similarly both on and off the training data. This objective arises from the observation that a hypothesis which fits the training data well but behaves erratically off the training set is not likely to generalize well. To detect erratic behavior we compare a hypothesis' behavior on the labeled training data to its behavior on unlabeled data. Specifically, we measure the distance that the hypothesis exhibits to a fixed "origin" function ϕ (chosen arbitrarily) on both data sets. If a hypothesis is behaving erratically off the labeled training set then it is likely that these distances will disagree. This effect is demonstrated in Figure 1 for two large degree polynomials that fit the training data well, but differ dramatically in their true error and their differences between on and off training set distance to a simple origin function.

To formulate a concrete training objective we propose the following measures: empirical training error plus an additive penalty

$$\hat{d}(h, \mathrm{P}_{Y|X}) + \left| d(h, \phi) - \hat{d}(h, \phi) \right|$$

and empirical error times a multiplicative penalty

$$\hat{d}(h, \mathrm{P}_{Y|X}) \times \max\left(\frac{d(h, \phi)}{\hat{d}(h, \phi)}, \frac{\hat{d}(h, \phi)}{d(h, \phi)} \right) \qquad (4)$$



$$\begin{aligned} \hat{d}(h, \mathrm{P}_{Y|X}) &= 0.004 \\ d(h, \mathrm{P}_{Y|X}) &= 19.31 \\ \hat{d}(g, \mathrm{P}_{Y|X}) &= 0.101 \\ d(g, \mathrm{P}_{Y|X}) &= 0.543 \\ \\ \hat{d}(h, \phi) &= 1.014 \\ d(h, \phi) &= 19.24 \\ \hat{d}(g, \phi) &= 1.010 \\ d(g, \phi) &= 0.928 \end{aligned}$$

Figure 1. Over-fitting effects of polynomial curve fitting.

In each case we compare the behavior of a candidate hypothesis h to the fixed origin ϕ. Thus, in either case, we seek to minimize empirical training error $\hat{d}(h, \mathrm{P}_{Y|X})$ plus (or times) a penalty that measures the discrepancy between the distance to the origin on the labeled training data and the origin distance on unlabeled data. Somewhat surprisingly, we have found that the *multiplicative* objective (4) generally performs much better, as it more harshly penalizes discrepancies between on and off training set behavior. Therefore, this is the form that we adopt below. Note that this penalty works in both directions: hypotheses that are much further from the origin on the training data than off are penalized strongly, but so are hypotheses that are significantly *closer* to the origin on the training data than off. The rationale behind this symmetric criterion is that both types of erratic behavior indicate that the observed training error is likely to be an unrepresentative reflection of the hypothesis' true error.

Although the max ratio penalty in (4) appears to be ad hoc, this objective is not entirely unprincipled. One nice property it has is that if the origin function ϕ happens to be the target conditional distribution $\mathrm{P}_{Y|X}$ then minimizing (4) becomes equivalent to minimizing true prediction error $d(h, \mathrm{P}_{Y|X})$. This is easy to see because when $\hat{d}(h, \mathrm{P}_{Y|X}) \leq d(h, \mathrm{P}_{Y|X})$ (which is almost certain) the criterion becomes $\hat{d}(h, \mathrm{P}_{Y|X}) d(h, \mathrm{P}_{Y|X}) / \hat{d}(h, \mathrm{P}_{Y|X}) = d(h, \mathrm{P}_{Y|X})$, and otherwise when $\hat{d}(h, \mathrm{P}_{Y|X}) > d(h, \mathrm{P}_{Y|X})$ we obtain $\hat{d}(h, \mathrm{P}_{Y|X})^2 / d(h, \mathrm{P}_{Y|X}) > d(h, \mathrm{P}_{Y|X})$ and are minimizing an upper bound on $d(h, \mathrm{P}_{Y|X})$. Note that this property would not hold for naively smoothed versions of this objective. Thus, minimizing (4) will result in near optimal generalization performance in this case. However, even if the origin does not exactly match the target, the objective still provably penalizes hypotheses that have small training error and large test error. To see this, note that for any hypothesis h

$$\frac{d(h, \phi)}{\hat{d}(h, \phi)} \;\geq\; \frac{d(h, \mathrm{P}_{Y|X}) - d(\phi, \mathrm{P}_{Y|X})}{\hat{d}(h, \mathrm{P}_{Y|X}) + \hat{d}(\phi, \mathrm{P}_{Y|X})} \qquad (5)$$

by the triangle inequality. Since ϕ and $\mathrm{P}_{Y|X}$ are not optimized on the training set we can expect $\hat{d}(\phi, \mathrm{P}_{Y|X}) \approx d(\phi, \mathrm{P}_{Y|X})$ for moderate sample sizes. Thus, (5) shows that if $\hat{d}(h, \mathrm{P}_{Y|X})$ is small (say, less than $d(\phi, \mathrm{P}_{Y|X})$) and $d(h, \mathrm{P}_{Y|X})$ is large (greater than $k \times d(\phi, \mathrm{P}_{Y|X})$, $k \geq 3$), then h's training error must be penalized by a significant ratio (at least $\frac{k-1}{2}$). By contrast, an alternative hypothesis g that achieves comparable training error and yet exhibits balanced behavior on and off the labeled training set (i.e., such that $\hat{d}(g, \mathrm{P}_{Y|X}) \approx d(g, \mathrm{P}_{Y|X})$) will be strongly preferred; in fact, such a g cannot over-fit by the same amount as h without violating (5). Importantly, the Bayes optimal hypothesis h^* will also tend to have $\hat{d}(h^*, \mathrm{P}_{Y|X}) \approx d(h^*, \mathrm{P}_{Y|X})$ and $\hat{d}(h^*, \phi) \approx d(h^*, \phi)$ since it too does not depend on the training set. Thus, h^* will typically achieve a small value of the objective, which will force any hypothesis that has a large over-fitting error (relative to $d(\phi, \mathrm{P}_{Y|X})$) to exhibit an objective value greater than the minimum.

Note that the sensitivity of the lower bound clearly depends on the distance between the origin and the target. If the origin is too far from the target then the lower bound is weakened and the criterion (4) becomes less sensitive to over-fitting. However, our experiments show that the objective is not unduly sensitive to the choice of ϕ, so long as is not too far from the data. In fact, even simple constant functions generally suffice.[3]

The outcome is a new regularization procedure that uses the training objective (4) to penalize hypotheses based on the given training data and on the unlabeled data. Therefore, in effect, the resulting procedure uses the unlabeled data to automatically set the level of regularization for a given problem. One goal of this research is to apply the new training objective to various hypothesis classes and see if it regularizes effectively across different data sets. We demonstrate this for several classes below. However, the regularization behavior is even subtler: since the penalization factor in (4) also depends on the *specific* labeled training set under consideration, the resulting procedure regularizes in a *data dependent* way. That is, the procedure adapts the penalization to the particular set of observed data. This raises the possibility of outperforming any regularization scheme that keeps a fixed penalization level across different training samples drawn from the same problem. In fact, we demonstrate below that such an improvement can be achieved in realistic hypothesis classes on real data sets.

[3]It is conceivable to reduce the dependence on a single origin function by considering a set of origin functions $\phi_1, ..., \phi_n$ and penalizing according to the maximum ratio. However, we have not observed any significant improvements in doing this and therefore drop the idea here.

4. Polynomial Regression

The first supervised learning task we consider is regression. Here $Y = \mathbb{R}$ and we measure prediction errors by the squared loss $err(\hat{y}, y) = (\hat{y} - y)^2$, yielding the distance measures based on (2). The regularizer introduced in this paper turns out to perform very well in such problems. In this case, our training objective can be expressed as choosing a hypothesis to minimize

$$\sum_{i=1}^{t}(h(x_i) - y_i)^2/t \quad \times$$
$$\max\left(\frac{\sum_{j=1}^{r}(h(x_j)-\phi(x_j))^2/r}{\sum_{i=1}^{t}(h(x_i)-\phi(x_i))^2/t}, \frac{\sum_{i=1}^{t}(h(x_i)-\phi(x_i))^2/t}{\sum_{j=1}^{r}(h(x_j)-\phi(x_j))^2/r}\right)$$

where $\{\langle x_i, y_i\rangle\}_{i=1}^{t}$ is the set of labeled training data, $\{\langle x_j\rangle\}_{j=1}^{r}$ is a set of unlabeled examples, and ϕ is a fixed origin (which we usually just set to be the constant function at the mean of the y labels). Note again that this training objective seeks hypotheses that fit the training data while simultaneously behaving similarly on the labeled and unlabeled data. The regularization effect with this multiplicative penalty is strong and yet surprisingly responsive to good hypotheses.

To give a first concrete demonstration of the new training objective we consider the problem of polynomial regression. Here $X = \mathbb{R}$ and we attempt to learn a polynomial predictor $h : X \to Y$. Thus, H is the class of polynomials, which can be naturally stratified into the subclasses of polynomials of degree $d = 1, 2, ...$ It is well known that fitting data with polynomials can lead to dramatic over-fitting, as shown in Figure 1.

To test the basic effectiveness of our approach, we ran a series of experiments that considered different target functions with varying smoothness characteristics. The idea was to arrange a series of conditions where polynomials would face varying degrees of difficulty in modeling the underlying regularities in the domain.

The first class of methods we compared against were *model selection* methods, which take the best fit polynomials of degree 0, 1, 2, ..., etc., and attempt to select the best one using various forms of complexity penalization and hold-out testing. The methods we compared were: 10-fold cross validation, CVT (Efron, 1979); structural risk minimization, SRM (Vapnik, 1996; Cherkassky et al., 1996); GCV (Craven & Wahba, 1979); AIC (Akaike, 1974); BIC (Schwarz, 1978); FPE (Shibata, 1981); CP (Mallows, 1973); RIC (Foster & George, 1994); and the metric based model selection strategy, ADJ, introduced in (Schuurmans, 1997). However, since none of the statistical methods, GCV, BIC, FPE, CP and RIC, performed competitively in our experiments, we report results only for GCV which performed the best among them.

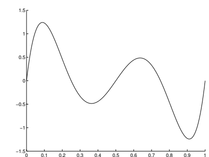
The second class of methods we compared against were *regularization* methods, which consider polynomials of maximum degree but penalize individual polynomials based on the size of their coefficients or their smoothness properties. The specific methods we considered were: a standard form of "ridge" penalization (or weight decay) which places a penalty $\lambda \sum_k a_k^2$ on polynomial coefficients a_k (Cherkassky & Mulier, 1998), and Bayesian maximum a posteriori inference with zero mean Gaussian priors on polynomial coefficients a_k with diagonal covariance matrix λI (MacKay, 1992; Young, 1977). Both of these methods require a regularization parameter λ to be set by hand. We refer to these methods as REG and MAP respectively.

To test the ability of our technique to automatically set the regularization level we tried a range of (fourteen) regularization parameters λ for the fixed regularization methods REG and MAP. For comparison purposes, we also report the results of the oracle regularizers, REG* and MAP*, that select the best λ value for each training set. Our experiments were conducted by randomly generating training points uniformly in the unit interval $[0, 1]$, labeling them using a target function, adding Gaussian noise, and then generating independent unlabeled points from $[0, 1]$.

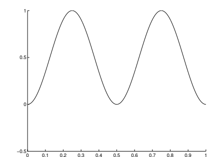
Table 1 shows the results of four different test problems at specified sample sizes and noise levels. The regularization criterion based on minimizing (4) is listed as ADA in our figures (for "adaptive" regularization). We tested ADA using different origin functions $\phi = \text{mean}\, y, \max y, 2 \max y, 4 \max y, 8 \max y$ to examine its robustness to ϕ. The results are quite strong. The first observation is that the model selection methods did not fare as well as the regularization techniques on these problems. Here model selection seems prone to making catastrophic over-fitting errors from time to time, whereas the regularization techniques appear to retain robust control. Interestingly, even the trusted 10-fold cross validation procedure CVT did not fare well in our experiments. The only model selection strategy to perform consistently is the metric-based method ADJ that also exploits unlabeled data.

The new adaptive regularization scheme ADA performed the best in all of our experiments. Table 1 shows that it outperforms fixed regularization strategies for all fixed choices of regularization parameter λ, even though the optimal choice varies across problems. This demonstrates that ADA is able to effectively tune its penalization behavior to the problem at hand. Moreover, since it outperforms even the best choice of λ for each data set, ADA also demonstrates the ability to adapt its penalization behavior to the specific train-

Table 1. A sampling of results from our polynomial experiments showing testing error (distance). Training sample size 20, unlabeled sample size 50, noise level 0.05. Results based on 1000 repetitions. ADA uses $\phi = \text{mean}\, y, 2 \max y, ..., 8 \max y$. REG uses $\lambda = 10^{-9}, ..., 50$.

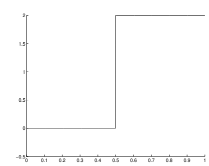


target = poly

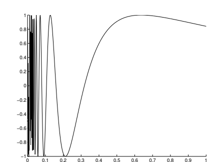


target = $\sin^2$

	target = poly			*target* = $\sin^2$		
	avg	med	std	avg	med	std
ADA	0.077	0.060	0.090	0.107	0.081	0.066
$2 \max$	0.073	0.059	0.054	0.137	0.083	0.168
$4 \max$	0.072	0.059	0.056	0.157	0.084	0.273
$8 \max$	0.075	0.059	0.115	0.230	0.084	0.844
REG*	0.147	0.082	0.121	0.140	0.092	0.099
10^{-9}	0.753	0.099	2.850	0.964	0.115	3.850
10^{-7}	0.514	0.094	1.780	0.797	0.124	3.120
10^{-5}	0.440	0.118	1.330	0.660	0.159	2.370
10^{-3}	0.558	0.225	1.190	0.582	0.237	1.150
10^{-2}	0.524	0.360	0.539	0.446	0.212	0.940
10^{-1}	0.454	0.337	0.508	0.509	0.291	0.500
0.5	0.523	0.396	0.337	0.405	0.355	0.145
1.0	0.532	0.499	0.086	0.358	0.342	0.066
5.0	0.520	0.511	0.038	0.353	0.341	0.040
50	0.519	0.513	0.030	0.353	0.342	0.033
MAP*	0.460	0.352	0.511	0.496	0.232	0.983
ADJ	0.116	0.062	0.188	0.188	0.114	0.150
10CV	0.321	0.065	3.160	0.559	0.132	1.980
SRM	0.163	0.062	1.230	0.576	0.128	2.430
GCV	2421	0.072	4.2e4	4.8e3	0.227	5.6e4



target = step



target = $\sin 1/x$

	target = step			*target* = $\sin 1/x$		
	avg	med	std	avg	med	std
ADA	0.391	0.366	0.113	0.444	0.425	0.085
$2 \max$	0.460	0.355	0.319	0.495	0.436	0.171
$4 \max$	0.556	0.367	0.643	0.533	0.427	0.326
$8 \max$	0.596	0.369	1.004	0.591	0.426	0.639
REG*	0.371	0.355	0.071	0.429	0.424	0.041
10^{-9}	7.940	0.664	38.50	4.250	0.758	28.00
10^{-7}	3.930	0.469	13.10	3.250	0.588	28.50
10^{-5}	2.570	0.457	8.360	1.830	0.588	12.80
10^{-3}	1.050	0.388	2.620	0.774	0.489	1.560
10^{-2}	0.697	0.397	0.825	0.558	0.452	0.550
10^{-1}	0.529	0.407	0.480	0.514	0.464	0.156
0.5	0.495	0.416	0.243	0.488	0.459	0.104
1.0	0.483	0.468	0.048	0.484	0.473	0.040
5.0	0.512	0.498	0.050	0.494	0.485	0.032
50	0.554	0.541	0.042	0.509	0.502	0.029
MAP*	0.496	0.400	0.385	0.651	0.476	0.989
ADJ	0.458	0.466	0.112	0.712	0.504	0.752
10CV	14.90	0.420	340.0	2.410	0.516	14.20
SRM	29.00	0.510	311.0	29.40	0.781	469.0
GCV	3.2e5	51.9	3.1e6	1.4e5	11.3	2.6e6

ing set, not just the given problem. It is also clear that ADA is fairly robust to the choice of ϕ: moving ϕ to a distant constant origin (even up to eight times the max y value) does not completely damage its performance.

5. Radial Basis Function Regression

To test our method on a more realistic task, we considered the problem of regularizing radial basis function (RBF) networks for regression. RBF networks are a natural generalization of interpolation and spline fitting methods. Given a set of prototype centers $c_1, ..., c_k$, an RBF representation of a prediction function h is given by

$$h(x) \;=\; \sum_{i=1}^{k} w_i \, g\left(\frac{\|x - c_i\|}{\sigma}\right) \tag{6}$$

where $\|x - c_i\|$ is the Euclidean distance between x and center c_i, and g is a response function with width parameter σ. Here we use a standard local (Gaussian) basis function $g(z) = e^{-z^2/\sigma^2}$.

Fitting with RBF networks is straightforward. The simplest approach is to place a prototype center on each training example and then determine the weight vector $\mathbf{w}$ that allows the network to fit the training y labels. This can be obtained by solving for $\mathbf{w}$ in[4]

$$\begin{bmatrix} g\left(\frac{\|x_1-x_1\|}{\sigma}\right) & \cdots & g\left(\frac{\|x_1-x_t\|}{\sigma}\right) \\ \vdots & & \vdots \\ g\left(\frac{\|x_t-x_1\|}{\sigma}\right) & \cdots & g\left(\frac{\|x_t-x_t\|}{\sigma}\right) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_t \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_t \end{bmatrix}$$

Although natural, exactly fitting data with RBF networks has the problem that the training data is generally over-fit in the process of replicating the y-labels. Many approaches exist for regularizing RBF networks. However, these techniques are often hard to apply because they involve setting various free parameters or controlling complex methods for choosing prototype centers, etc. (Cherkassky & Mulier, 1998; Bishop, 1995). The simplest regularization approaches are to add a ridge penalty to the weight vector, and minimize

$$\sum_{i=1}^{t} (h(x_i) - y_i)^2 + \lambda \sum_{i=1}^{t} w_i^2 \tag{7}$$

where h is given as in (6) (Cherkassky & Mulier, 1998). An alternative approach is to add a non-parametric penalty on curvature (Poggio & Girosi, 1990), but the resulting procedure is similar. To apply these methods in practice one has to make an intelligent choice of the width parameter σ and the regularization parameter λ. Unfortunately, these choices interact, and it is often hard to set them by hand without extensive visualization and experimentation with the data set.

[4] This solution is guaranteed to exist and be unique for distinct training points and most natural basis functions g, including the Gaussian basis used here.

In this section we investigate how effectively the ADA regularizer is able to automatically select the width parameter σ and regularization parameter λ in an RBF network on real regression problems. Here again the basic idea is to use unlabeled data to make these choices automatically and adaptively. We compare ADA to a large number of ridge regularization procedures, each corresponding to the penalty (7) with different fixed choices of σ and λ (thirty five in total).

In this study we experimented with a number of regression problems from the StatLib (lib.stat.cmu.edu) and UCI (www.ics.uci.edu/~mlearn/MLRepository.html) machine learning repositories. In our experiments, a data set was randomly split into a training, unlabeled, and test set, and then each of the methods was run on this split. We repeated the random splits 100 times to obtain our results. Table 2 shows that ADA regularization is able to choose width and regularization parameters that achieve effective generalization performance across a range of data sets. Here ADA performs better than any fixed regularizer on every problem except one, and moreover it even beats the oracle regularizer REG* in all but two problems. This shows that the adaptive criterion is not only effective at choosing good regularization parameters for a given problem, it can choose them adaptively based on the given training data to yield improvements over fixed regularizers.

6. Conditional Density Estimation

The approach we have introduced in this paper is by no means restricted to regression problems. In fact, as mentioned in Section 3, the generic regularization criterion (4) can be applied to a wide range of supervised learning tasks, including classification and conditional density estimation. Here we present a short demonstration of our method for regularizing conditional probability models.

Consider a setting where hypotheses make probabilistic predictions, $h(x) \in [0,1]$, of y labels in $\{0,1\}$. Here we use the log loss to measure prediction error, $err(\hat{y}, y) = -y \log \hat{y} - (1-y) \log(1-\hat{y})$. In this situation we measure "distances" between conditional probability models using the KL-divergence. The goal is to minimize the true log loss of the hypothesis h, which amounts to minimizing the KL-divergence between h and the target conditional density, $d(h \| \mathrm{P}_{Y|X})$.

Table 2. RBF results showing test errors (distances) averaged over 100 splits of the dataset. Standard deviation are given for ADA and REG*.

AAUP data set

ADA 0.0197 ± 0.004 | REG* 0.0361 ± 0.009

REG	λ=0.0	0.5	1.0	5.0	10
σ=0.01	0.0393	0.0512	0.0545	0.0597	0.0617
0.05	0.0460	0.0507	0.0528	0.0562	0.0582
0.10	0.0498	0.0501	0.0517	0.0547	0.0571
0.25	0.0551	0.0488	0.0500	0.0531	0.0561
0.50	0.0592	0.0478	0.0488	0.0522	0.0559
0.75	0.0617	0.0472	0.0481	0.0519	0.0561
1.00	0.0641	0.0467	0.0477	0.0517	0.0563

ABALONE data set

ADA 0.034 ± 0.0046 | REG* 0.045 ± 0.0055

REG	λ=0.0	0.5	1.0	5.0	10
σ=0.01	0.1717	0.0527	0.0537	0.0565	0.0592
0.05	0.2131	0.0517	0.0521	0.0546	0.0580
0.10	0.2986	0.0511	0.0515	0.0541	0.0580
0.25	0.3999	0.0504	0.0507	0.0539	0.0585
0.50	0.9654	0.0500	0.0503	0.0540	0.0594
0.75	0.9073	0.0498	0.0502	0.0542	0.0601
1.00	0.4030	0.0497	0.0501	0.0545	0.0607

BODYFAT data set

ADA 0.131 ± 0.0171 | REG* 0.129 ± 0.0150

REG	λ=0.0	0.5	1.0	5.0	10
σ=0.01	0.1521	0.1351	0.1366	0.1418	0.1467
0.05	0.1588	0.1339	0.1356	0.1420	0.1479
0.10	0.1621	0.1336	0.1354	0.1426	0.1489
0.25	0.1667	0.1334	0.1353	0.1437	0.1507
0.50	0.1704	0.1334	0.1354	0.1448	0.1523
0.75	0.1726	0.1334	0.1355	0.1457	0.1533
1.00	0.1742	0.1335	0.1356	0.1463	0.1541

BOSTON-C data set

ADA 0.150 ± 0.0212 | REG* 0.155 ± 0.0197

REG	λ=0.0	0.5	1.0	5.0	10
σ=0.01	0.1626	0.1673	0.1702	0.1782	0.1820
0.05	0.1631	0.1667	0.1696	0.1774	0.1808
0.10	0.1643	0.1667	0.1696	0.1774	0.1807
0.25	0.1668	0.1669	0.1699	0.1777	0.1809
0.50	0.1698	0.1672	0.1703	0.1781	0.1813
0.75	0.1719	0.1674	0.1705	0.1784	0.1816
1.00	0.1737	0.1676	0.1707	0.1786	0.1818

STRIKES data set

ADA 0.0249 ± 0.0068 | REG* 0.0185 ± 0.0058

REG	λ=0.0	0.5	1.0	5.0	10
σ=0.01	0.0456	0.0964	0.1090	0.1439	0.1563
0.05	0.0334	0.0933	0.1091	0.1483	0.1600
0.10	0.0291	0.0936	0.1109	0.1510	0.1620
0.25	0.0243	0.0955	0.1148	0.1548	0.1647
0.50	0.0212	0.0983	0.1189	0.1578	0.1667
0.75	0.0196	0.1006	0.1217	0.1595	0.1679
1.00	0.0185	0.1024	0.1239	0.1607	0.1686

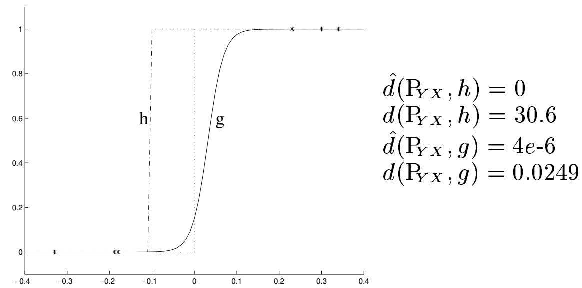


Figure 2. Maximum likelihood (h) versus ADA-regularized (g) logistic regression.

A well known problem with using log loss for conditional density estimation is that even trivial probability models can over-fit training data. That is, a simple conditional probability model can get a small negative log likelihood on the training data

$$\hat{d}(h\|\mathrm{P}_{Y|X}) \;=\; \frac{1}{t}\sum_{i=1}^{t} -y_i \log h(x_i) - (1-y_i)\log(1-h(x_i))$$

and yet still get large log loss errors on test examples.

To illustrate, consider the example of one dimensional logistic regression where hypotheses are defined by

$$h(x) \;=\; \frac{1}{1+e^{-ax-b}}$$

for parameters a and b. This defines a "smoothed" step function with a crossover value at $x = -\frac{b}{a}$ (the point where $h(x) = 1 - h(x) = \frac{1}{2}$). Here, given data $\langle x_1, y_1\rangle, ..., \langle x_t, y_t\rangle$, the naive maximum likelihood approach will attempt to choose parameters a and b that minimizes $\hat{d}(h\|\mathrm{P}_{Y|X})$. However, if the data happens to be separable, as shown in Figure 2, then zero training error can be achieved by pushing a and b to infinite values while maintaining any split point $-\frac{b}{a}$ between the two classes of y-labels. If, however, this split point does not correspond exactly to a hard split in the target conditional $\mathrm{P}_{Y|X}$ (and there is no reason to believe that it would) the resulting hypothesis h, which makes hard classifications, will obtain huge (approaching infinite) errors on any misclassified test example.

We can avoid this type of over-fitting simply by regularizing the hypothesis h. Doing this with our adaptive regularization scheme is straightforward. Figure 2 shows the results of minimizing the penalized objective

$$\hat{d}(h\|\mathrm{P}_{Y|X}) \times \max\left(\frac{d(h\|\phi)}{\hat{d}(h\|\phi)}, \frac{\hat{d}(h\|\phi)}{d(h\|\phi)}\right)$$

using the origin function $\phi(x) = \frac{1}{2}$. Here we obtained a smoothed hypothesis that obtains similar training and test errors—unlike the naive maximum likelihood hypothesis that badly over-fits on these problems.

Table 3. Some decision tree pruning results on UCI repository data sets, showing size and test error over 100 splits.

	un-pruned		C4.5 pruned		ADA-pruned	
	size	test	size	test	size	test
random	120	50.5	105	50.5	51	50.2
optdigit	269	15.3	250	15.2	234	15.2
iris	7	8.9	6	8.8	6	9.3
glass	11	10.8	11	10.8	10	12.8
ecoli	32	24.1	22	22.4	22	23.6
vote	21	6.7	8	5.2	14	6.9
crx	56	19.8	28	18.0	23	17.3
soybean	146	19.7	75	17.5	124	19.7
hypo	25	0.94	19	0.83	23	0.87

7. Classification

Finally, we briefly note that the regularization approach developed in this paper can be easily applied to classification problems. In this situation, Y is usually a small discrete set and we measure prediction error by the misclassification loss, $err(\hat{y}, y) = 1_{(\hat{y} \neq y)}$. Here, distances are measured by the disagreement probability $d(f, g) = \mathrm{P}_X(f(x) \neq g(x))$. Our generic regularization objective (4) could be directly applied in this setting. However, our preliminary results (on decision tree pruning) are not strong, and it appears that our technique will not work as decisively for classification problems as it does for regression and conditional density estimation problems; see Table 3.

However, we believe that the difficulty has an intuitive explanation: Since classification functions are essentially histogram-like, they limit the ability of our methods to detect erratic behavior off the labeled training sample. This is because histograms, being flat across large regions, tend to behave similarly in large neighborhoods around training points—to the extent that distances on labeled and unlabeled data points are often very similar, even for complex histograms. Coping with this apparent limitation in our approach remains grounds for future research.

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Automat. Control*, *19*, 716–723.

Bishop, C. (1995). *Neural networks for pattern recognition.* Oxford: Clarendon Press.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–40.

Cherkassky, V., & Mulier, F. (1998). *Learning from data.* New York: Wiley.

Cherkassky, V., Mulier, F., & Vapnik, V. (1996). Comparison of VC-method with classical methods for model selection. Preprint.

Craven, P., & Wahba, G. (1979).Smoothing noisy data with spline functions. *Numer. Math.*, *31*, 377–403.

Efron, B. (1979). Computers and the theory of statistics. *SIAM Review*, *21*, 460–480.

Foster, D., & George, E. (1994). The risk inflation criterion for multiple regression. *Ann. Statist.*, *22*, 1947–1975.

Freund, Y., & Schapire, R. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *J. Comput. System Sci.*, *55*, 119–139.

Krogh, A., & Vedelsby, J. (1995). Neural network ensembles, cross validation, and active learning. *Advances in Neural Information Processing Systems 7* (pp. 231–238).

MacKay, D. (1992). Bayesian interpolation. *Neural Computation*, *4*, 415–447.

Mallows, C. (1973). Some comments on C_p. *Technometrics*, *15*, 661–676.

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

Ripley, B. (1996). *Pattern recognition and neural networks.* Cambridge: Cambridge University Press.

Rissanen, J. (1986). Stochastic complexity and modeling. *Ann. Statist.*, *14*, 1080–1100.

Schuurmans, D. (1997). A new metric-based approach to model selection. *Proceedings of National Conference on Artificial Intelligence* (pp. 552–558).

Schuurmans, D., Ungar, L., & Foster, D. (1997). Characterizing the generalization performance of model selection strategies. *Proceedings of International Conference on Machine Learning* (pp. 340–348).

Schwarz, G. (1978). Estimating the dimension of a model. *Ann. Statist.*, *6*, 461–464.

Shibata, R. (1981). An optimal selection of regression variables. *Biometrika*, *68*, 45–54.

Vapnik, V. (1996). *The nature of statistical learning theory.* New York: Springer-Verlag.

Young, A. (1977). A Bayesian approach to prediction using polynomials. *Biometrika*, *64*, 309–317.