Appears in
Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97),
Providence, RI, July 1997.

A New Metric-Based Approach to Model Selection

Dale Schuurmans

Institute for Research in Cognitive Science
University of Pennsylvania
Philadelphia, PA 19104-6228
daes@linc.cis.upenn.edu

NEC Research Institute
4 Independence Way
Princeton, NJ 08540
dale@research.nj.nec.com

Abstract

We introduce a new approach to model selection that performs better than the standard complexity-penalization and hold-out error estimation techniques in many cases. The basic idea is to exploit the intrinsic metric structure of a hypothesis space, as determined by the natural distribution of *unlabeled* training patterns, and use this metric as a reference to detect whether the empirical error estimates derived from a small (labeled) training sample can be trusted in the region around an empirically optimal hypothesis. Using simple metric intuitions we develop new geometric strategies for detecting overfitting and performing robust yet responsive model selection in spaces of candidate functions. These new metric-based strategies dramatically outperform previous approaches in experimental studies of classical polynomial curve fitting. Moreover, the technique is simple, efficient, and can be applied to most function learning tasks. The only requirement is access to an auxiliary collection of unlabeled training data.

Introduction

In the standard problem of learning a prediction function $h : X \to Y$ from training examples $\langle x_1, y_1\rangle$, ..., $\langle x_t, y_t\rangle$, the idea is to take the small set of y-labels and extend them to a total prediction function h over the entire domain X. Our goal is to produce a function h that predicts the y-labels of future (possibly unseen) x's as accurately as possible, where we measure the accuracy of our predictions by some specified error function $err(\hat{y}, y)$.[1]

The simplest prototypical approach to this problem is to first conjecture a suitable class of hypothesis functions H (*e.g.*, by specifying a neural net architecture, or some other representation class), and then choose the hypothesis $h^* \in H$ that minimizes the empirical error $\sum_{j=1}^{t} err(h^*(x_j), y_j)$ on the training set. Of course, the key to making this approach work is to choose the right hypothesis class H. One could argue for example that it would be advantageous to make H as expressive as possible, since this would afford us the greatest chance of representing a reasonable hypothesis. However, by making H *too* expressive we run the risk of "overfitting" the training data and producing a hypothesis function that predicts poorly on future test examples (see, *e.g.*, Figure 3 below). In fact, there is a well-developed statistical theory which supports this intuition by saying that for h^* to be reliably near the best function in H we require a training sample size that is proportional to the "complexity" of the hypothesis class H (Vapnik 1982; Pollard 1984; Haussler 1992). This suggests that we must restrict the complexity of our hypothesis class somehow. Of course, this can introduce the opposite problem of *underfitting*. That is, we might restrict H so severely as to eliminate any reasonable hypotheses, even when perfectly acceptable prediction functions exist. This, then, is the fundamental dilemma of machine learning: we need to make our hypothesis classes as expressive as possible to maximize our chances representing a good hypothesis, but we also need to restrict these classes to ensure that we can reliably distinguish good from bad hypotheses (Geman, Bienenstock, & Doursat 1992). Thus, there is a tradeoff between our ability to represent a good function and our ability to identify a good function, if one exists. The question of what to do in the face of this dilemma dominates much of machine learning research.

Most successful applied machine learning systems employ some sort of mechanism to balance between hypothesis complexity and data-fit. Perhaps the most common strategy for coping with this dilemma in practice is to use some form of automatic *Model Selection*: First stratify the hypothesis class into a sequence of nested classes $H_0 \subset H_1 \subset ... = H$, and then (somehow) choose a class which has the appropriate complexity

[1]Note that most research on function learning considers specific representations for domain objects x (*e.g.*, attribute vectors), range predictions y (*e.g.*, binary or real valued label), and prediction functions h (*e.g.*, feedforward neural networks or decision trees); and focuses on specific error functions like zero-one loss $1_{(\hat{y} \neq y)}$ or squared error $\|\hat{y} - y\|^2$. Here we will take a simple abstract view that encompasses most such choices.

for the given training data. To understand how we might make this choice, note that for a given training set $\langle x_1, y_1\rangle, ..., \langle x_t, y_t\rangle$ we obtain a corresponding sequence of empirically optimal functions $h_0^*, h_1^*, ...$ from each successive subclass (Figure 1). The basic model selection problem is to choose one of these functions on the basis of their observed empirical errors. Note that these errors are monotonically decreasing, and therefore choosing the function with minimum training error simply leads to choosing a function from the largest class. Therefore the trick is to invoke some other criteria beyond empirical error minimization to make this choice.

Previous approaches Currently, two model selection strategies predominate. The most common strategy is *complexity-penalization*. Here we first assign increasing complexity values $c_0, c_1, ...$ to the successive function classes, and then choose a hypothesis from $h_1^*, h_2^*, ..$ that minimizes some prior combination of complexity and empirical error (*e.g.*, the additive combination $c_i + \lambda err(h_i^*)$). There are many variants of this basic approach, including the minimum description length principle (Rissanen 1986), "Bayesian" maximum a posteriori selection, structural risk minimization (Vapnik 1982; 1996), "generalized" cross validation (Craven & Wahba 1979) (different from real cross validation; below), and regularization (Moody 1992). These strategies differ in the specific complexity values they assign and the particular tradeoff function they optimize, but the basic idea is the same.

The other most common strategy is *hold-out testing*. Here we ask for the given set of training data which hypothesis class H_i generalizes best? We answer this by partitioning the training set $1, ..., t$ into a pseudo-training set $1, ..., k$ and a hold-out test set $k+1, ..., t$, and then use the pseudo-training set to obtain a sequence of pseudo-hypotheses $\hat{h}_0, \hat{h}_1, ...,$ *etc.* We then use the hold-out test set to obtain an *unbiased* estimate of the true errors of these pseudo-hypotheses. (Note that the training set errors tend to be gross underestimates in general.) From these unbiased estimates, we can simply choose the hypothesis class H_i that yields the pseudo-hypothesis $\hat{h}_i$ with the smallest estimated error. Once H_i has been selected, we return the function $h_i^* \in H_i$ that obtains minimum empirical error on the *entire* training sequence. Again, there are many variants of this basic strategy—having to do with repeating the pseudo-train pseudo-test split many times and averaging the results to choose the final hypothesis class; *e.g.*, 10-fold cross validation, leave-one-out testing, bootstrapping, *etc.* (Efron 1979; Weiss & Kulikowski 1991). Repeated testing in this manner does introduce some bias in the error estimates, but the results are still generally better than considering a single hold-out partition (Weiss & Kulikowski 1991).

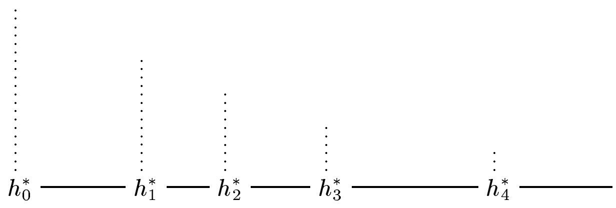

Figure 1: Sequence of empirically optimal functions determined by a stratification $H_0 \subset H_1 \subset ...$ and training set. Dotted lines indicate decreasing empirical errors.

Idea

Here we propose a fundamentally different approach to model selection that seems to work better than either complexity-penalization or hold-out testing in many cases. Our basic idea is to exploit the intrinsic geometry of the function learning task which arises from a simple statistical model of the problem: Assume the training and test examples are independent random observations drawn from a joint distribution P_{XY} on $X \times Y$. Then we can decompose this distribution into the conditional distribution of Y given X, $P_{Y|X}$, and the marginal distribution P_X on X. Note that when learning a prediction function $h : X \to Y$ we are really only interested in approximating the conditional $P_{Y|X}$. However our approach will be to try and exploit knowledge about P_X to help us make better decisions about which hypothesis h to choose. In fact, for now, we will assume that we actually *know* P_X and see how far this gets us.[2]

How can knowing P_X help? Well, the first thing it does is give us a natural measure of the "distance" between two hypotheses h and g. In fact, we can obtain a natural (pseudo) *metric* on the space of hypotheses via the definition $d(h, g) =$ "$\int_X err(h(x), g(x))\, dP_X$"; that is, we measure the distance between two functions by their average discrepancy on random x-objects (the reason for the quotes is explained below). Moreover, we can extend this definition to include the target conditional $P_{Y|X}$ via the definition $d(h, P_{Y|X}) =$ "$\int_X \int_Y err(h(x), y)\, dP_{Y|x} dP_X$"; which means that we can interpret the true error of a function h as the *distance* between h and the target object $P_{Y|X}$. Importantly, these definitions are compatible in the sense that the defined metric d satisfies the standard axioms over $H \cup \{P_{Y|X}\}$.

Notice how this now gives us a nice geometric view of the problem (Figure 2): We have a nested sequence of spaces $H_0 \subset H_1 \subset ...$, each with a closest function $h_0, h_1, ...$ to the target $P_{Y|X}$, where the distances are decreasing. However, we do not

[2] We will note below that any information we require about P_X can be obtained from *unlabeled* training examples. Thus, the key leverage of our approach will be based on having access to a collection of unlabeled data on which we can stabilize model selection behavior.

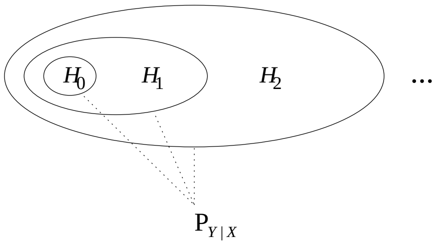

Figure 2: Geometric view of model selection: given a nested sequence of *spaces* $H_0 \subset H_1 \subset ...$, try to find the closest function to $P_{Y|X}$, using estimated distances.

get to observe these real distances. Rather, we are given a training sample and have to choose a hypothesis from the sequence of *empirically* closest functions $h_0^*, h_1^*, ...,$ which have decreasing estimates $\widehat{d(h_i^*, P_{Y|X})} = $ "$\sum_{j=1}^t err(h_i^*(x_j), y_j)/t$" (Figure 1). The key point though is that we now have more information at our disposal: not only do we have estimated distances to $P_{Y|X}$, we now know the true distances *between* functions in the sequence! (*E.g.*, as indicated by the bold lines in Figure 1.)

Our idea is to somehow use this additional information to choose a better hypothesis. One intuition is that these inter-function distances can help us detect overfitting. For example, consider two hypotheses h_i^* and h_{i+1}^* that both have small estimated distances to $P_{Y|X}$ and yet have a large true distance between them. We claim that one should worry about choosing the second function. Why? Well, if the true distance between h_i^* and h_{i+1}^* is large, then both functions cannot be close to $P_{Y|X}$, by simple geometry. This means then that one of the estimates must be wrong, and we know to trust the earlier estimate more than the latter. In fact, if both $\widehat{d(h_i^*, P_{Y|X})}$ and $\widehat{d(h_{i+1}^*, P_{Y|X})}$ really were accurate estimates, they would have to satisfy the triangle inequality with the known distance $d(h_i^*, h_{i+1}^*)$; *i.e.*,

$$\widehat{d(h_i^*, P_{Y|X})} + \widehat{d(h_{i+1}^*, P_{Y|X})} \geq d(h_i^*, h_{i+1}^*). \qquad (1)$$

Since these empirical distances eventually become gross underestimates in general (because the h_i^* are explicitly chosen to minimize the empirical distance on the training set) we can use the triangle inequality test to detect when these estimates become untrustworthy. In fact, this forms the basis of a very simple model selection strategy (TRI): choose the last function in the sequence $h_1^*, h_2^*, ...$ that does not violate the triangle inequality with any preceding function. This simple procedure turns out to work surprisingly well in experimental situations.

Case study: Polynomial curve fitting

To explore the effectiveness of our simple model selection procedure we considered the classical problem of fitting a polynomial to a set of points (Figure 3). Specifically, we considered a function learning problem where $X = I\!R$, $Y = I\!R$, and the goal is to minimize the squared prediction error, $err(\hat{y}, y) = (\hat{y} - y)^2$. Here we considered polynomial hypotheses $h : I\!R \to I\!R$ under the natural stratification $H_0 \subset H_1 \subset ...$ into polynomials of degree $0, 1, ...,$ *etc.* The motivation for studying this task is that it is a classical well-studied problem, that still attracts a lot of interest (Galarza, Rietman, & Vapnik 1996; Cherkassky, Mulier, & Vapnik 1996; Vapnik 1996). Moreover, polynomials create a difficult model selection problem that has a strong tendency to produce catastrophic overfitting effects (Figure 3). Another benefit is that polynomials are an interesting and nontrivial class for which there are efficient techniques for computing best fit hypotheses.

To apply our metric strategy TRI to this task we need to define a suitable metric distance d under the presumed distribution P_X. For the squared error measure we can define the distance between two functions by $d(h, g) = \left(\int_X (h(x) - g(x))^2 \, dP_X\right)^{1/2}$ and the distance to $P_{Y|X}$ by $d(h, P_{Y|X}) = \left(\int_X \int_Y (h(x) - y)^2 \, dP_{Y|x} dP_X\right)^{1/2}$. This establishes a verifiable (pseudo) metric over $H \cup \{P_{Y|X}\}$. (Notice that we have to take square roots to get a metric here; hence the earlier need for quotes.) Also, for a given training set $\langle x_1, y_1\rangle, ..., \langle x_t, y_t\rangle$, we can define the corresponding empirical distance estimate by $\widehat{d(h, P_{Y|X})} = \left(\sum_{j=1}^t (h(x_j) - y_j)^2/t\right)^{1/2}$.

To determine the efficacy of TRI we compared its performance to a number of standard model selection strategies, including two well-known penalization strategies—generalized cross validation GCV (Craven & Wahba 1979) and structural risk minimization SRM (Vapnik 1996) (under the formulations reported in (Cherkassky, Mulier, & Vapnik 1996))—and 10-fold cross validation 10CV, a standard hold-out method (Efron 1979; Kohavi 1995).

We conducted a simple series of experiments by fixing a uniform domain distribution P_X on the unit interval $[0, 1]$, and then fixing various target functions $f : [0, 1] \to I\!R$. To generate training samples we first drew a sequence of values, $x_1, ..., x_t$, computed the target function values $f(x_1), ..., f(x_t)$, and added independent Gaussian noise to each, to obtain the labeled training sequence $\langle x_1, y_1\rangle, ..., \langle x_t, y_t\rangle$. For a given training sample we then computed the series of best fit polynomials $h_0^*, h_1^*, ...$ of degree $0, 1, ...,$ *etc.* Given this sequence, each model selection strategy will choose some hypothesis h_i^* on the basis of the observed empirical errors. To implement TRI we assumed that it had access to the *known* uniform distribution P_X over $[0, 1]$ in order to compute the true distances between polynomials in the sequence. (We return to the issue of estimating P_X below.)

Our main emphasis in these experiments was to minimize the true distance between the final hypothesis and the target conditional $P_{Y|X}$. That is, we are primarily concerned with choosing a hypothesis that ob-

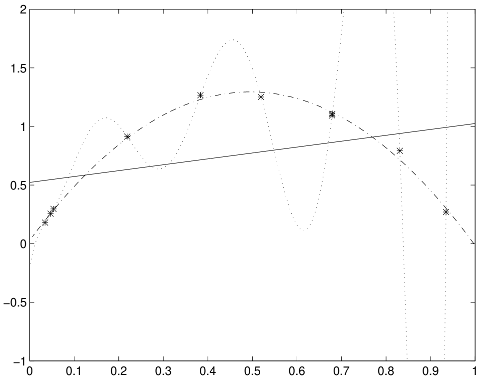

Figure 3: Minimum squared error polynomials of degrees 1, 2, and 9 for a set of 10 training points. Notice that the high degree polynomial demonstrates erratic behavior off the training set.

tains a small prediction error on future test examples, independent of its complexity level.[3] To determine the effectiveness of the various selection strategies, we measured the *ratio* of the true error of the polynomial they selected to the best true error among polynomials in the sequence $h_0^*, h_1^*, \ldots$. The rationale for doing this is that we wish to measure the model selection strategy's ability to approximate the best hypothesis in the given sequence—not find a better function from outside the sequence.[4]

Experiment 1 Tables 1–3 show the results obtained for fitting a step function $f(x) = \text{step}(x \geq 0.5)$ corrupted by Gaussian noise. (The strategy ADJ in the tables is explained below.) We obtained these results by repeatedly generating training samples of a fixed size and recording the approximation ratio achieved by each strategy. These tables record the distribution of ratios produced by each strategy for training sample sizes of 10, 20 and 30 respectively, given 800 trials each. The results are quite dramatic. TRI achieved median approximation ratios of 1.03, 1.07 and 1.06 for training sample sizes 10, 20 and 30 respectively. This compares favorably to the median approximation ratios 1.24, 1.34 and 2.14 achieved by SRM, and 1.24, 1.16 and 1.16 achieved by 10CV (GCV was dramatically

[3]This is not the only criteria one could imagine optimizing here. For example, one could be interested in finding a simple model of the underlying phenomenon that gives some insight into its fundamental nature, rather than simply producing a function that predicts well on future test examples (Heckerman & Chickering 1996). However, we will focus on the traditional machine learning goal of minimizing prediction error.

[4]One could consider more elaborate strategies that choose hypotheses from outside the sequence; *e.g.*, by averaging several hypotheses together (Opitz & Shavlik 1996; Breiman 1994). However, we will not pursue this idea here.

	percentiles of approximation ratios				
method	25	50	75	95	100
TRI	1.00	1.03	1.17	1.44	2.42
10CV	1.07	1.24	1.51	7.38	854.3
SRM	1.05	1.24	1.44	4.24	58.3
GCV	1.76	10.6	98.7	2399	4.8×10^5
ADJ	1.00	1.16	1.40	3.50	152.5

Table 1: Fitting $f(x) = \text{step}(x \geq 0.5)$ with $P_X = U(0,1)$ and $\sigma = 0.05$. Distribution of approximation ratios achieved at training sample size $t = 10$, showing percentiles of approximation ratios achieved after 800 repeated trials.

	percentiles of approximation ratios				
method	25	50	75	95	100
TRI	1.00	1.07	1.19	1.44	2.18
10CV	1.06	1.16	1.39	4.60	1482
SRM	1.13	1.34	2.65	40.98	13,240
GCV	1.64	27.0	895.0	1×10^5	2×10^7
ADJ	1.03	1.13	1.25	1.58	3.42

Table 2: Same as Table 1 but with $t = 20$ examples.

	percentiles of approximation ratios				
method	25	50	75	95	100
TRI	1.00	1.06	1.17	1.42	2.02
10CV	1.06	1.16	1.37	6.22	58.9
SRM	1.17	2.14	22.0	1894	3.2×10^6
GCV	4.20	73.0	1233	46,504	4.3×10^8
ADJ	1.06	1.15	1.27	1.53	2.08

Table 3: Same as Table 1 but with $t = 30$ examples. *I.e.*, fitting $f(x) = \text{step}(x \geq 0.5)$ with $P_X = U(0,1)$ and $\sigma = 0.05$, given $t = 30$ training examples; showing percentiles of approximation ratios achieved after 800 trials.

	percentiles of approximation ratios			
method	25	50	75	95
TRI	1.00	1.06	1.17	1.40
10CV	1.01	1.31	4.15	65.9
SRM	9.57	1820	9.5×10^6	1.1×10^{10}
GCV	25.2	3×10^6	5.9×10^{11}	8.3×10^{16}
ADJ	1.00	1.00	1.10	1.20

Table 4: Same at Table 3 but with $P_X = N(0.5, 1)$.

	percentiles of approximation ratios				
method	25	50	75	95	100
TRI	1.02	1.12	1.28	1.59	2.44
10CV	1.06	1.16	1.38	10.3	163.7
SRM	1.40	5.03	35.5	459.4	5.2×10^5
GCV	2.20	11.9	105.6	4138	1.6×10^7
ADJ	1.08	1.17	1.28	1.82	4.93

Table 5: Same as Table 3 but with $f(x) = \sin(1/x)$.

method	percentiles of approximation ratios 25	50	75	95	100
TRI	1.30	2.00	3.35	5.48	15.1
10CV	1.03	1.32	1.85	7.04	82.5
SRM	1.03	1.29	1.83	5.34	3978
GCV	1.04	1.41	2.93	37.0	1.9×10^5
ADJ	1.02	1.31	1.88	4.19	8.92

Table 6: Same as Table 3 but with $f(x) = \sin^2(2\pi x)$.

worse on these trials).[5] However, the striking difference was TRI's *robustness* against overfitting. In fact, although the penalization strategy SRM performed reasonably well fairly often, it was prone to making *catastrophic* overfitting errors. Even the normally well-behaved cross-validation strategy 10CV made significant overfitting errors from time to time. This is evidenced by the fact that in 800 trials with a training sample of size 30 (Table 3) TRI produced a *maximum* approximation ratio of 2.02, whereas 10CV produced a worst case approximation ratio of 59, and the penalization strategies SRM and GCV produced worst case ratios of 3×10^6 and 4×10^8 respectively! (The 95th percentiles were TRI 1.42, 10CV 6.22, SRM 1894, GCV 4.6×10^4.) In fact, TRI's robustness against overfitting is not a surprise: one can *prove* that TRI cannot produce an approximation ratio greater than 3, if we assume that *(i)* TRI makes it to the best hypothesis h^* in the sequence, and *(ii)* the empirical error of h^* is an underestimate. (The proof is by simple geometry and is given in the appendix.)

The basic flavor of these results remains unchanged at different noise levels and for different domain distributions P_X. In fact, much stronger results are obtained for wider tailed domain distributions like Gaussian (Table 4), and "difficult" target functions like $\sin(1/x)$ (Table 5). Here SRM and GCV can be forced into a regime of constant catastrophe, 10CV noticeably degrades, and yet TRI retains the same performance levels shown in Table 3.

Experiment 2 Of course, a step function is a rather pathological target to fit with a polynomial, and therefore it is important to consider other more "natural" targets which might be better suited to polynomial approximation. In fact, by repeating the previous experiments with a more benign target function $f(x) = \sin^2(2\pi x)$ we obtain quite different results. Table 6 shows that procedure TRI does not fare as well in this case—obtaining median approximation ratios of 1.4, 2.1 and 2.0 for training sample sizes 10, 20 and 30 respectively (compared to 1.6, 1.24 and 1.29 for SRM, and 1.7, 1.4 and 1.3 for 10CV). A close examination of the data reveals that the reason for this performance

[5] Although the penalization strategies appear to be performing worse for larger training sample sizes, their performance improves again at sample sizes greater than 100.

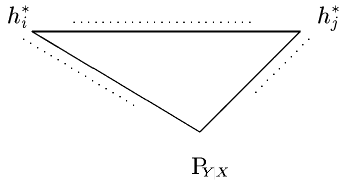

Figure 4: The real and estimated distances between successive hypotheses h_i and h_j and the target $P_{Y|X}$. Solid lines indicate real distances, dashed lines indicate empirical distance estimates.

drop is that TRI systematically gets stuck at lower degree polynomials. In fact, there is a simple geometric explanation for this: the even-degree polynomials (after 4) all give reasonable fits to $\sin^2(2\pi x)$ whereas the odd-degree fits have a tail in the wrong direction. This creates a huge distance between successive polynomials, and causes the triangles to break between the even and odd degree fits, even when the large even-degree polynomial is a good approximation. So although the metric-based TRI strategy is strongly robust against overfitting, it can be prone to systematic underfitting in seemingly benign cases. (Similar results were obtained for other polynomial and polynomial-like target functions.) This problem leads us to consider a reformulated procedure.

Strategy 2: Adjusted distance estimates

The final idea we explore is based on the observation that we are in fact dealing with two metrics here: the true metric d defined by the joint distribution P_{XY} and an empirical metric $\hat{d}$ determined by the labeled training sequence. Now given these two metrics, consider the triangle formed by two hypotheses h_i^* and h_j^* and the target conditional $P_{Y|X}$ (Figure 4). Notice that there are six distances involved—three real and three estimated, of which the true distances to $P_{Y|X}$ are the only two we care about, and yet these are the only two we don't have! The key observation though is that the real and estimated distances between hypotheses h_i^* and h_j^*, $d(h_i^*, h_j^*)$ and $\widehat{d(h_i^*, h_j^*)}$, give us an *observable* relationship between d and $\hat{d}$ in the local vicinity. In fact, we can adopt the naive assumption that observed relationship between h_i^* and h_j^* also holds between h_j^* and $P_{Y|X}$. Note that if this were the case, we would obtain a better estimate of $d(h_j^*, P_{Y|X})$ simply by adjusting the training set distance $\widehat{d(h_j^*, P_{Y|X})}$ according to the observed ratio $d(h_i^*, h_j^*)/\widehat{d(h_i^*, h_j^*)}$.[6] In fact, adopting this as a simple heuristic leads to a surprisingly effective model selection procedure (ADJ): given the hypothesis sequence $h_1^*, h_2^*, \ldots$, first multiply each

[6] Note that since we expect $\hat{d}$ to be an underestimate in general, we expect this ratio to be typically larger than 1.

estimated distance $\widehat{d(h_j^*, P_{Y|X})}$ by the largest observed ratio $d(h_i^*, h_j^*)/\widehat{d(h_i^*, h_j^*)}$, $i < j$, and then choose the function in the sequence that has the smallest *adjusted* distance estimate to $P_{Y|X}$. This simple but motivated procedure seems to overcome the underfitting problems associated with TRI while still retaining TRI's robustness against overfitting.

To demonstrate the efficacy of ADJ we repeated the previous experiments including ADJ as a new competitor. Our results show that ADJ robustly outperformed the standard complexity-penalization and hold-out methods in all cases considered—spanning a wide variety of target functions, noise levels, and distributions P_X. Although space limitations preclude a complete and systematic exposition of our results, Tables 1–6 demonstrate typical outcomes. In particular, Table 6 shows that ADJ avoids the underfitting problems that plague TRI; it responsively selects high order approximations when this is supported by the data. Moreover, Table 3 shows that ADJ is still extremely robust against overfitting, even in situations where the standard approaches make catastrophic errors. Although the results reported here are anecdotal, our full suite of experiments strongly suggest that ADJ outperforms standard techniques across a wide variety of polynomial regression problems. Overall, this is the best model selection strategy we have observed for these polynomial regression tasks.

Estimating P_X

Of course, one can always argue that these results are not terribly useful since the metric-based strategies TRI and ADJ both require knowledge of the true domain distribution P_X. This is clearly an unreasonable assumption in practice. However, it is trivial to observe that we can obtain information about P_X from *unlabeled* training instances. In fact, many important function learning applications have large collections of unlabeled training data available (*e.g.*, image, speech and text databases), so these metric-based techniques could still apply to a wide range of practical situations—provided they are robust to using only *estimated* distances. To explore this issue, we repeated our previous experiments, but gave TRI and ADJ only a small reference sample to estimate inter-hypothesis distances. We found that these strategies were in fact extremely robust to using approximate distances. Table 7 shows that as few as 100 unlabeled examples (just over three times the number of labeled examples) were still sufficient for TRI and ADJ to perform nearly as well as before. Moreover, Table 7 shows that these techniques only begin to significantly break down once we consider fewer unlabeled than labeled training examples. Although the evidence is anecdotal, this robustness was observed across a wide range of problems. It remains an important direction for future research to systematically characterize the range of reference sample sizes for which this holds.

	percentiles of approximation ratios				
method	25	50	75	95	100
$\widehat{\text{TRI}}$(100)	1.00	1.07	1.18	1.81	7.07
$\widehat{\text{TRI}}$ (50)	1.01	1.10	1.28	2.56	23.3
$\widehat{\text{TRI}}$ (25)	1.03	1.14	1.51	7.89	2342
$\widehat{\text{ADJ}}$(100)	1.06	1.15	1.27	1.56	3.50
$\widehat{\text{ADJ}}$ (50)	1.05	1.14	1.29	1.69	10.4
$\widehat{\text{ADJ}}$ (25)	1.06	1.16	1.34	2.13	158.0

Table 7: Same as Table 3 but using a small number of *unlabeled* examples (in parentheses) to estimate P_X.

Finally, note that this still yields a reasonably efficient model selection procedure, since computing inter-hypothesis distances involves making only a single pass down the reference list of unlabeled examples. This is a strong advantage over standard hold-out techniques like 10CV which repeatedly call the hypothesis generating mechanism to generate pseudo-hypotheses—an extremely expensive operation in many applications.

Conclusion

We have introduced a new approach to the classical model selection problem that is based on exploiting the intrinsic geometry of the function learning task. These new techniques significantly outperform standard approaches in a wide range of polynomial regression tasks. The primary source of this advantage is that our metric-based strategies are able to detect dangerous situations and avoid making catastrophic overfitting errors, while still being responsive enough to adopt complex models when this is supported by the data. They accomplish this by attending to the real distances between hypotheses. Note that complexity-penalization strategies completely ignore this information, and as a result are heavily punished in our experiments. Hold-out methods implicitly take some of this information into account, but do so indirectly and less effectively than the metric-based strategies introduced here. Although there is no "free lunch" in general (Schaffer 1994) and we cannot claim to obtain a universal improvement for every model selection problem (Schaffer 1993), we claim that one should be able to exploit additional information about the task (here knowledge of P_X) to obtain significant improvements across a wide range of problem types and conditions. Our empirical results for polynomial regression support this view.

An important direction for future research is to develop theoretical support for our strategies. Some progress in this direction is reported in a companion paper (Schuurmans, Ungar, & Foster 1997) which develops a general characterization of the difficulty of model selection problems based on the standard bias/variance decomposition of expected hypothesis error (Geman, Bienenstock, & Doursat 1992). Here we characterize model selection problems by the shapes of their approximation-error and variance profiles, and use this to delineate the conditions where traditional techniques are most prone to catastrophic mistakes and where our techniques obtain their greatest advantage.

It remains open as to whether TRI and ADJ are the best possible ways to exploit the hypothesis distances afforded by P_X. We plan to investigate alternative strategies which might be more effective in this regard.

Finally we note that there is nothing about our approach that is specific to polynomial curve fitting! The techniques developed here can easily be applied to other hypothesis classes familiar to AI research; including neural networks, radial basis functions, and decision trees. In fact, our metric-based approach easily generalizes to *classification* learning tasks as well, since the classification loss function $err(\hat{y}, y) = 1_{\{\hat{y} \neq y\}}$ directly gives a metric via the definitions $d(h, g) = P_X(h(x) \neq g(x))$ and $d(h, P_{Y|X}) = P_{XY}(h(x) \neq y)$. However, as discussed in (Schuurmans, Ungar, & Foster 1997), we do not expect to achieve as dramatic successes here, since classification involves a bounded loss which does not permit catastrophic errors (*i.e.*, distances greater than 1). Nevertheless, applying our techniques to classification tasks is another important direction for future research. Here we hope to compare our results with the earlier study (Kearns *et al.* 1995).

Acknowledgements

Much of this work was performed at the National Research Council Canada. I would like to thank Rob Holte, Joel Martin and Peter Turney for their help in developing the nascent ideas, and Adam Grove, Lyle Ungar, Dean Foster, Geoff Hinton and Rob Tibshirani for their later insightful comments. I would also like to thank Vladimir Vapnik for suggesting polynomial curve fitting as a suitable test problem, and for providing several useful references. IRCS, under the generous hand of Aravind Joshi, paid for the extra page.

Appendix

We prove that TRI cannot exhibit an approximation ratio larger than 3 if we assume that *(i)* TRI makes it to the best hypothesis h_i^* in the sequence, and *(ii)* the empirical error of h_i^* is an underestimate. Consider a hypothesis h_j^* which follows h_i^* in the sequence, and assume $d(h_j^*, P_{Y|X}) > 3d(h_i^*, P_{Y|X})$. We show that h_j^* must fail the triangle test (1) with h_i^*: First, notice that h_j^*'s error and the triangle inequality imply that $3d(h_i^*, P_{Y|X}) < d(h_j^*, P_{Y|X}) \leq d(h_i^*, P_{Y|X}) + d(h_i^*, h_j^*)$; and hence $d(h_i^*, h_j^*) > 2d(h_i^*, P_{Y|X})$. But now recall that $d(\widehat{h_j^*, P_{Y|X}}) \leq d(\widehat{h_i^*, P_{Y|X}})$ for $j > i$, and also, by assumption, $d(\widehat{h_i^*, P_{Y|X}}) < d(h_i^*, P_{Y|X})$. Therefore we have $d(h_i^*, h_j^*) > 2d(h_i^*, P_{Y|X}) > d(\widehat{h_i^*, P_{Y|X}}) + d(\widehat{h_j^*, P_{Y|X}})$; which contradicts (1). Thus, TRI will not consider h_j^*.

References

Breiman, L. 1994. Bagging predictors. Technical Report 421, Statistics Department, UC Berkeley.

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

Craven, P., and 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.

Galarza, C.; Rietman, E.; and Vapnik, V. 1996. Applications of model selection techniques to polynomial approximation. Preprint.

Geman, S.; Bienenstock, E.; and Doursat, R. 1992. Neural networks and the bias/variance dilemma. *Neural Comp.* 4:1–58.

Haussler, D. 1992. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Infor. Comput.* 100:78–150.

Heckerman, D., and Chickering, D. 1996. A comparison of scientific and engineering criteria for Bayesian model selection. Technical Report MSR-TR-96-12, Microsoft Research.

Kearns, M.; Mansour, Y.; Ng, A.; and Ron, D. 1995. An experimental and theoretical comparison of model selection methods. In *Proceedings COLT-95*.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings IJCAI-95*.

Moody, J. 1992. The effective number of parameters: An analysis of generalization and regularization in nonlinear learning systems. In *Proceedings NIPS-4*.

Opitz, D., and Shavlik, J. 1996. Generating accurate and diverse members of a neural-network ensemble. In *Proceedings NIPS-8*.

Pollard, D. 1984. *Convergence of Stochastic Processes*. New York: Springer-Verlag.

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

Schaffer, C. 1993. Overfitting avoidance as bias. *Mach. Learn.* 10(2):153–178.

Schaffer, C. 1994. A conservation law for generalization performance. In *Proceedings ML-94*.

Schuurmans, D.; Ungar, L.; and Foster, D. 1997. Characterizing the generalization performance of model selection strategies. In *Proceedings ML-97*. To appear.

Vapnik, V. 1982. *Estimation of Dependences Based on Empirical Data*. New York: Springer-Verlag.

Vapnik, V. 1996. *The Nature of Statistical Learning Theory*. New York: Springer-Verlag.

Weiss, S. M., and Kulikowski, C. A. 1991. *Computer Systems that Learn*. San Mateo: Morgan Kaufmann.