24 Metric-Based Approaches for Semi-Supervised Regression and Classification

Dale Schuurmans
Finnegan Southey
Dana Wilkinson
Yuhong Guo

Semi-supervised learning methods typically require an explicit relationship to be asserted between the labeled and unlabeled data—as illustrated, for example, by the neighbourhoods used in graph-based methods. Semi-supervised model selection and regularization methods are presented here that instead require only that the labeled and unlabeled data are drawn from the same distribution. From this assumption, a metric can be constructed over hypotheses based on their predictions for unlabeled data. This metric can then be used to detect untrustworthy training error estimates, leading to model selection strategies that select the richest hypothesis class while providing theoretical guarantees against over-fitting. This general approach is then adapted to regularization for supervised regression and supervised classification with probabilistic classifiers. The regularization adapts not only to the hypothesis class but also to the specific data sample provided, allowing for better performance than regularizers that account only for class complexity.

24.1 Introduction

The tradeoff between over-fitting and under-fitting is a fundamental dilemma in machine learning and statistics. Given a collection of data points $\mathbf{x} \in X$, each associated with a dependent value $y \in Y$, one often wishes to learn a function or hypothesis which effectively predicts the correct y given any $\mathbf{x}$. If a hypothesis is chosen from a class that is too complex for the data, there is a good chance it will exhibit large test error even though its training error is small—i.e., over-fitting the training data. This occurs 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—thus diminishing the ability to distinguish good hypotheses from bad. Since significantly different hypotheses cannot be simultaneously accurate, one must restrict the set of hypotheses to be able to reliably differentiate between accurate and inaccurate predictors. On the other hand, selecting hypotheses from an overly restricted class can prevent one from being able to express a good approximation to the ideal predictor, thereby causing important structure in the training data to be ignored—i.e., under-fitting the training data. Since both under-fitting and over-fitting result in large test error, they must be avoided simultaneously. Consequently, a popular research topic in learning is to find *automated* methods for calibrating hypothesis complexity. The work presented here exploits unlabeled data in a novel fashion to achieve this goal.

We consider two classical approaches to this problem, typically referred to as *model selection* and *regularization* respectively [CM98, Vap96, Vap98]. In *model selection* one first takes a base hypothesis class, H, decomposes it into a discrete collection of subclasses $H_0 \subset H_1 \subset \cdots = H$ (say, organized in a nested chain, or lattice) and then, given training data, attempts to identify the optimal subclass from which to choose the final hypothesis.[1] 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 [Ris86] and various statistical selection criteria [FG94]); and *hold-out testing* (e.g., cross-validation and bootstrapping [Efr79]). *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 [CM98, Rip96, Bis95]) or penalizes their global smoothness properties (e.g., minimizing curvature [PG90]). 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.

The contribution presented here is the derivation of *parameter-free* methods for model selection and regularization that improve on the robustness of standard approaches by using unlabeled data. As has been seen in other sections of the book, most semi-supervised learning techniques require explicit assumptions about the relationship between labeled and unlabeled data. For the methods presented here, the only assumption required is that the labeled data and the unlabeled data come from the same distribution. The methods we propose automatically differentiate hypotheses based on the difference of their behaviour off of the labeled training set (i.e., behaviour at points not covered by the training set). Like many of the semi-supervised learning approaches proposed in this book (e.g., Chapter 10), our methods regularize in a data-specific fashion rather than simply penalizing model complexity. This allows modern techniques to potentially outperform traditional fixed regularizers that penalize complexity identically across different training samples.

[1] The term *model selection* has also been used to refer to other processes in machine learning and statistics, such as choosing the kernel for support vector machines or Bayesian model selection, but we restrict our attention to the classical form described above.

To begin, Section 24.2 introduces the idea of metric spaces for hypotheses, allowing the geometric characterization of the supervised learning problem. Section 24.3 investigates how unlabeled data can be used to perform *model selection* in nested sequences of hypothesis spaces. The strategies developed are shown to experimentally outperform standard model selection methods and have been proved to be robust in theory. Section 24.4 considers *regularization* and shows how the proposed model selection strategies can be extended to a generalized training objective for supervised regression. Here the idea is to use unlabeled data to automatically tune the degree of regularization for a given task without having to set free parameters by hand. The resulting regularization technique adapts its behaviour to a given training set and can outperform standard fixed regularizers for a given problem. Section 24.5 extends the earlier regression approach from Section 24.4 to probabilistic classifiers. Finally, Section 24.6 concludes with an examination of potential avenues for future research.

24.2 Metric structure of supervised learning

In supervised learning, one takes a sequence of training pairs $\langle \mathbf{x}_1, y_1 \rangle, ..., \langle \mathbf{x}_l, y_l \rangle$ and attempts to infer a hypothesis function $h : X \to Y$ that achieves small prediction error *err*$(h(\mathbf{x}), y)$ on future test examples. This basic paradigm covers many of the tasks studied in machine learning research.

For model selection and regularization tasks it is necessary to be able to compare hypothesis functions. The approach we pursue in this chapter is to exploit a concrete notion of distance between hypothesis functions. Consider the metric structure on a space of hypothesis functions that arises from a simple statistical model of the supervised learning problem: Assume the examples $\langle \mathbf{x}, y \rangle$ are generated by a fixed joint distribution P_{XY} on $X \times Y$. In learning a hypothesis function $h : X \to Y$ the primary interest is in modeling some aspect of the conditional distribution $\mathrm{P}_{Y|X}$. Here the utility of using extra information about the marginal domain distribution P_X to choose a good hypothesis is investigated. Note that information about P_X can be obtained from a collection of *un*labeled training examples $\mathbf{x}_{l+1}, ..., \mathbf{x}_n$. The significance of having information about the domain distribution P_X is that it defines a natural *(pseudo) metric* on the space of hypotheses. That is, for any two hypothesis functions f and g, one can obtain a measure of the distance between them by computing the expected disagreement in their predictions

$$d(f, g) \quad \triangleq \quad \varphi\left(\int \textit{err}(f(\mathbf{x}), g(\mathbf{x}))\, d\mathrm{P}_X\right) \tag{24.1}$$

where *err*$(\hat{y}, y)$ is the natural measure of prediction error for the problem at hand (e.g., regression or classification) and φ is an associated normalization function that recovers the standard metric axioms.

For the problem of regression, prediction error can be measured by squared difference *err*$(\hat{y}, y) = (\hat{y} - y)^2$ or some similar loss. For classification problems, prediction error can be measured with the misclassification loss *err*$(\hat{y}, y) = 1_{(\hat{y} \neq y)}$. The standard metric properties to be satisfied are non-negativity $d(f, g) \geq 0$, symmetry $d(f, g) = d(g, f)$, and the triangle inequality $d(f,g) \leq d(f,h) + d(h,g)$. It turns out that most typical prediction error functions admit a metric of this type.

For example, in regression the distance between two prediction functions can be measured by:

$$d(f,g) \quad = \quad \left(\int (f(\mathbf{x}) - g(\mathbf{x}))^2 \, d\mathrm{P}_X \right)^{1/2}$$

where the normalization function $\varphi(z) = z^{1/2}$ establishes the metric properties. In classification, the distance between two classifiers can be measured by:

$$\begin{aligned} d(f,g) \quad &= \quad \int 1_{(f(\mathbf{x}) \neq g(\mathbf{x}))} \, d\mathrm{P}_X \\ &= \quad \mathrm{P}_X(f(\mathbf{x}) \neq g(\mathbf{x})) \end{aligned}$$

where no normalization is required to achieve a metric. Importantly, these definitions can be generalized to include the target *conditional distribution* in an analogous manner:

$$d(\mathrm{P}_{Y|X}, h) \quad \triangleq \quad \varphi\left(\int\!\!\int \mathit{err}(h(\mathbf{x}), y) \, d\mathrm{P}_{Y|x} \, d\mathrm{P}_X \right) \tag{24.2}$$

That is, one can interpret the true error of a hypothesis function h with respect to a target conditional $\mathrm{P}_{Y|X}$ as a *distance* between h and $\mathrm{P}_{Y|X}$. The significance of this definition is that it is consistent with the previous definition (Equation 24.1) and one can therefore embed the entire supervised learning problem in a common metric space structure.

To illustrate: in regression, Equation 24.2 yields the root mean squared error of a hypothesis:

$$d(\mathrm{P}_{Y|X}, h) \quad = \quad \left(\int\!\!\int (h(\mathbf{x}) - y)^2 \, d\mathrm{P}_{Y|x} \, d\mathrm{P}_X \right)^{1/2}$$

and in classification it gives the true misclassification probability:

$$\begin{aligned} d(\mathrm{P}_{Y|X}, h) \quad &= \quad \int\!\!\int 1_{(h(\mathbf{x}) \neq y)} \, d\mathrm{P}_{Y|x} \, d\mathrm{P}_X \\ &= \quad \mathrm{P}_{XY}(h(\mathbf{x}) \neq y) \end{aligned}$$

Together, the definitions in Equations 24.1 and 24.2 show how to impose a global metric space view of the supervised learning problem (Figure 24.1). Given labeled training examples $\langle \mathbf{x}_1, y_1 \rangle, ..., \langle \mathbf{x}_l, y_l \rangle$, the goal is to find the hypothesis h in a space H that is closest to a target conditional $\mathrm{P}_{Y|X}$ under the distance measure (Equation 24.2). If there is also a large set of u auxiliary unlabeled examples $\mathbf{x}_{l+1}, ..., \mathbf{x}_n$, such that $u = n - l$, then one can also accurately estimate the distances between alternative hypotheses f and g within H, effectively giving Equation 24.1:

$$\tilde{d}(f,g) \quad \triangleq \quad \varphi\left(\frac{1}{u} \sum_{j=l+1}^{n} \mathit{err}(f(\mathbf{x}_j), g(\mathbf{x}_j)) \right) \tag{24.3}$$

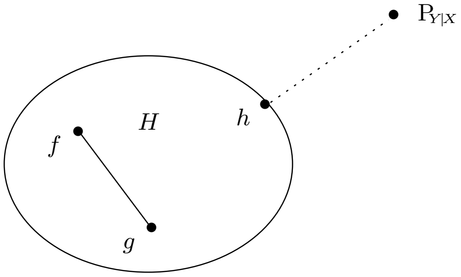


Figure 24.1 Metric space view of supervised learning: Unlabeled data can accurately estimate distances between functions f and g within H, however only limited labeled data is available to estimate the closest function h to $P_{Y|X}$.

That is, for sufficiently large u, the distances defined in Equation 24.3 will be very close to the distances defined in Equation 24.1. In fact, below we sill generally assume that u is large enough to ensure $\tilde{d}(f,g) \approx d(f,g)$. However, the distances between hypotheses and the target conditional $P_{Y|X}$ (Equation 24.2) can only be weakly estimated using the (presumably much smaller) set of labeled training data:

$$\hat{d}(P_{Y|X}, h) \quad \stackrel{\triangle}{=} \quad \varphi\left(\frac{1}{l}\sum_{i=1}^{l} err(h(\mathbf{x}_i), y_i)\right) \tag{24.4}$$

This measure need not be close to Equation 24.2. The challenge then is to approximate the closest hypothesis to the target conditional as accurately as possible using the available information (Equations 24.3 and 24.4) in place of the true distances (Equations 24.1 and 24.2).

This metric space perspective will be used to devise novel model selection and regularization strategies that exploit inter-hypothesis distances measured on an auxiliary set of unlabeled examples. The proposed approach is applicable to any supervised learning problem that admits a reasonable metric structure. In particular, all strategies will be expressed in terms of a generic distance measure that does not depend on other aspects of the problem.

24.3 Model selection

First consider the process of using *model selection* to choose the appropriate level of hypothesis complexity to fit to data. This is, conceptually, the simplest approach to automatic complexity control for supervised learning. The idea is to stratify the hypothesis class H into a sequence (or lattice) of nested subclasses $H_0 \subset H_1 \subset \cdots = H$, and then, given training data, somehow choose a class that has the proper complexity for the given data. To understand how one might make this choice, note that for a given training sample $\langle \mathbf{x}_1, y_1 \rangle, \ldots, \langle \mathbf{x}_l, y_l \rangle$ one can, in principle, obtain the corresponding sequence of empiri-

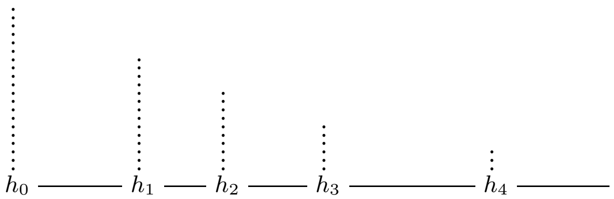


Figure 24.2 Sequence of empirically optimal functions induced by a chain $H_0 \subset H_1 \subset ...$ on a given training set: Dotted lines indicate decreasing optimal training distances $\hat{d}(h_0, \mathrm{P}_{Y|X})$, $\hat{d}(h_1, \mathrm{P}_{Y|X})$, ... and solid lines indicated distances *between* hypotheses. The final hypothesis must be selected on the basis of these estimates.

cally optimal functions $h_0 \in H_0$, $h_1 \in H_1, \ldots$

$$h_k \;=\; \arg\min_{h \in H_k} \quad \varphi\left(\frac{1}{l}\sum_{i=1}^{l} \mathit{err}(h(\mathbf{x}_i), y_i)\right) \;=\; \arg\min_{h \in H_k} \quad \hat{d}(\mathrm{P}_{Y|X}, h)$$

That is, here we assume an empirical risk minimization procedure is used to select a candidate function from each class, and moreover we assume a unique minimizer exists for each H_k.[2] The problem is to select one of these functions based on the observed training errors $\hat{d}(\mathrm{P}_{Y|X}, h_0)$, $\hat{d}(\mathrm{P}_{Y|X}, h_1), \ldots$ (Figure 24.2). However, because each hypothesis class subsumes those before it, these errors must monotonically decrease (assuming one can fully optimize in each class) and therefore choosing the function with smallest training error inevitably leads to over-fitting. Some other criterion beyond mere empirical-error minimization must be invoked to make the final selection.

As mentioned, two basic model selection strategies currently predominate: *complexity penalization* and *hold-out testing*. However, neither of these approaches attends to the metric distances between hypotheses, nor do they offer an obvious way to exploit auxiliary unlabeled data. By adopting the metric space view of Section 24.2, however, a useful new perspective on model selection can be obtained. In our setting, the chain $H_0 \subset H_1 \subset \cdots \subset H$ can be interpreted as a sequence of hypothesis *spaces* wherein one can measure the distance between candidate hypotheses using unlabeled data. Note that it is still not possible to directly measure the distances from hypotheses to the target conditional $\mathrm{P}_{Y|X}$ and therefore they must be estimated based on a small labeled training sample. However, the fact that there are distances *between* functions in the sequence can be exploited—this additional information being used to make a better choice (Figure 24.2).

24.3.1 Strategy 1: Triangle inequality

The first intuition explored is that inter-hypothesis distances can help detect over-fitting in a very simple manner. Consider two hypotheses h_k and h_{k+1} that both have a small estimated distance to $\mathrm{P}_{Y|X}$ and yet have a large true distance between them. In this situation their

[2] This uniqueness assumption is reasonable for regression problems but generally does not hold for classification problems under 0-1 loss; see Section 24.5 below.

Procedure TRI
- Given hypothesis sequence $h_0, h_1, ...$
- Choose the last hypothesis h_ℓ in the sequence that satisfies the triangle inequality, $\tilde{d}(h_k, h_\ell) \leq \hat{d}(h_k, \mathrm{P}_{Y|X}) + \hat{d}(\mathrm{P}_{Y|X}, h_\ell)$, with every preceding hypothesis h_k, $0 \leq k < \ell$.

Figure 24.3 Triangle inequality model selection procedure.

should be concern in selecting the second hypothesis, because if the true distance between h_k and h_{k+1} is indeed large then both functions cannot be simultaneously close to $\mathrm{P}_{Y|X}$, by simple geometry. This implies that at least one of the distance estimates to $\mathrm{P}_{Y|X}$ must be inaccurate. The earlier estimate should be more trusted because it comes from a more restricted class that is less likely to overfit. In fact, if both $\hat{d}(\mathrm{P}_{Y|X}, h_k)$ and $\hat{d}(\mathrm{P}_{Y|X}, h_{k+1})$ really were accurate estimates they would have to satisfy the ***triangle inequality*** with the known distance $d(h_k, h_{k+1})$; that is:

$$\hat{d}(\mathrm{P}_{Y|X}, h_k) + \hat{d}(\mathrm{P}_{Y|X}, h_{k+1}) \geq d(h_k, h_{k+1}) \tag{24.5}$$

Since these empirical distances eventually become significant underestimates in general (because a particular h_i is explicitly chosen to minimize the empirical distance on the labeled training set) the triangle inequality provides a useful test for detecting when these estimates become inaccurate. In fact, this basic test forms the basis of a simple model selection strategy, TRI (Figure 24.3), that works surprisingly well in many situations.

24.3.2 Example: Polynomial regression

To demonstrate this method (and all subsequent methods developed here), first consider the problem of polynomial curve fitting. This is a supervised learning problem where $X = \mathbb{R}$, $Y = \mathbb{R}$, and the goal is to minimize the squared prediction error, *err*$(\hat{y}, y) = (\hat{y} - y)^2$. Specifically, consider polynomial hypotheses $h : \mathbb{R} \to \mathbb{R}$ under the natural stratification $H_0 \subset H_1 \subset ...$ into polynomials of degree at most $0, 1, ...$, etc. The motivation for studying this task is that it is a well-studied problem that still attracts a lot of interest [CMV97, GRV96, Vap96, Vap98]. Moreover, polynomials create a difficult model selection problem that has a strong tendency to produce catastrophic over-fitting effects. Another benefit is that polynomials are an interesting and non-trivial class for which there are efficient techniques for computing best-fit hypotheses.

To apply the metric-based approach to this task, define the metric d in terms of the squared prediction error *err*$(\hat{y}, y) = (\hat{y} - y)^2$ with a square root normalization $\varphi(z) = z^{1/2}$, as discussed in Section 24.2. To evaluate the efficacy of TRI on this problem, its performance was compared to a number of standard model selection strategies, including structural risk minimization (SRM) [CMV97, Vap98], RIC [FG94], SMS [Shi81], GCV [CW79], BIC [Sch78], AIC [Aka74], CP [Mal73], and FPE [Aka70]. TRI was also compared to 10-fold cross validation, CVT (a standard hold-out method [Efr79, Koh95]).

A simple series of experiments was conducted by fixing a domain distribution P_X on $X = \mathbb{R}$ and then fixing various target functions $f : \mathbb{R} \to \mathbb{R}$. The specific target functions used in the experiments are shown in Figure 24.4. To generate training samples a sequence of values $(\mathbf{x}_1, \ldots, \mathbf{x}_l)$ were drawn, then the target function values $(f(\mathbf{x}_1), \ldots, f(\mathbf{x}_l)$

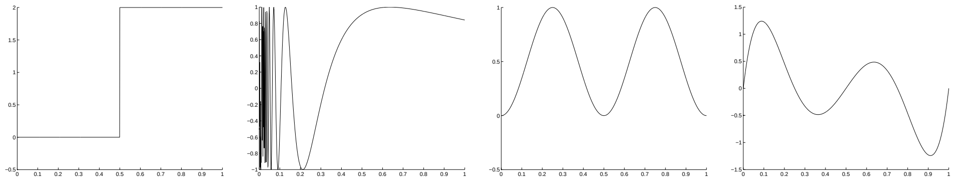


Figure 24.4 Target functions used in the polynomial curve fitting experiments (in order): step$(x \geq 0.5)$, $\sin(1/x)$, $\sin^2(2\pi x)$, and a fifth degree polynomial.

computed and perturbed by adding independent Gaussian noise with standard deviation $\sigma = 0.05$ to each. This resulted in a labeled training sequence $\langle \mathbf{x}_1, y_1 \rangle, \ldots, \langle \mathbf{x}_l, y_l \rangle$. For a given training sample the series of best fit polynomials $h_0, h_1, \ldots$ of degree $0, 1, \ldots$ was computed. Given this sequence, each model selection strategy will choose some hypothesis h_k on the basis of the observed empirical errors. The implementation of TRI was given access to u auxiliary unlabeled examples $\mathbf{x}_{l+1}, \ldots, \mathbf{x}_n$ in order to estimate the true distances between polynomials in the sequence.

The main emphasis in these experiments was to minimize the true distance between the final hypothesis and the target conditional $\mathrm{P}_{Y|X}$. That is, the primary concern was choosing a hypothesis that obtained a small prediction error on future test examples, independent of its complexity level. To determine the effectiveness of the various selection strategies, the *ratio* of the true error (distance) of the polynomial they selected to the best true error among polynomials in the sequence $h_0, h_1, ...$, was measured. This means that the optimum achievable ratio was 1. The rationale for doing this was 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.[3]

Table 24.1 shows the results obtained for approximating a step function $f(\mathbf{x}) =$ step$(\mathbf{x} \geq 0.5)$ corrupted by Gaussian noise, where the marginal distribution P_X is uniform on $[0, 1]$. The strategy ADJ in the tables is explained in Section 24.3.3 below. These results were obtained by repeatedly generating training samples of a fixed size and recording the approximation ratio achieved by each strategy. The tables record the distribution of ratios produced by each strategy for a training sample size of $l = 30$, using $u = 200$ unlabeled examples to measure inter-hypothesis distances, repeated over 1000 trials. The initial results appear to be quite positive. TRI achieves a median approximation ratio of 1.08. This compares favorably to the median approximation ratio 1.54 achieved by SRM, and 1.17 achieved by CVT. The remaining complexity penalization strategies—GCV, FPE, etc.—all performed significantly worse on these trials. However, the most notable difference was TRI's robustness against over-fitting. In fact, although the penalization strategy SRM performed reasonably well much of the time, it was prone to making periodic but catastrophic over-fitting errors. Even the normally well-behaved cross-validation strategy

[3] One could consider more elaborate strategies that choose hypotheses from outside the sequence; e.g., by averaging several hypotheses together [KV95, OS96, Bre96]. However, this idea will not be pursued further here.

$l=30$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	1.00	1.08	1.17	4.69	1.51	5.41	5.45	2.72	1.06
50	1.08	1.17	1.54	34.8	9.19	39.6	40.8	19.1	1.14
75	1.19	1.37	9.68	258	91.3	266	266	159	1.25
95	1.45	6.11	419	4.7e3	2.7e3	4.8e3	5.1e3	4.0e3	1.51
100	2.18	643	1.6e7	1.6e7	1.6e7	1.6e7	1.6e7	1.6e7	2.10

Table 24.1 Fitting $f(x) = \text{step}(x \geq 0.5)$ with $\mathrm{P}_X = U(0,1)$ and $\sigma = 0.05$. Table gives distribution of approximation ratios achieved at training sample size $l = 30$, showing percentiles of approximation ratios achieved in 1000 repeated trials.

$l=30$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	1.02	1.08	1.34	2.80	1.89	3.16	3.67	2.80	1.08
50	1.14	1.20	4.74	12.1	9.67	14.1	15.8	13.8	1.17
75	1.30	1.63	33.2	61.5	55.2	70.1	81.6	72.4	1.30
95	1.72	23.5	306	1.2e3	479	1.3e3	1.3e3	1.3e3	1.81
100	2.68	325	1.4e5	5.2e5	1.4e5	5.2e5	5.2e5	3.9e5	9.75

Table 24.2 Fitting $f(x) = \sin(1/x)$ with $\mathrm{P}_X = U(0,1)$ and $\sigma = 0.05$. Table gives distribution of approximation ratios achieved at training sample size $l = 30$, showing percentiles of approximation ratios achieved in 1000 repeated trials.

$l=30$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	1.50	1.00	1.00	1.00	1.00	1.00	1.00	1.02	1.01
50	3.51	1.16	1.03	1.05	1.11	1.02	1.08	1.45	1.27
75	4.15	1.64	1.45	1.48	2.02	1.39	1.88	6.44	1.60
95	5.51	5.21	5.06	4.21	26.4	5.01	19.9	295	3.02
100	9.75	124	1.4e3	20.0	9.1e3	28.4	9.4e3	1.0e4	8.35

Table 24.3 Fitting $f(x) = \sin^2(2\pi x)$ with $\mathrm{P}_X = U(0,1)$ and $\sigma = 0.05$. Table gives distribution of approximation ratios achieved at training sample size $l = 30$, showing percentiles of approximation ratios achieved in 1000 repeated trials.

$l=30$	TRI	CVT	SRM	RIC	GCV	BIC	AIC	FPE	ADJ
25	7.80	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
50	8.58	1.01	1.00	1.00	1.01	1.00	1.00	1.08	1.00
75	9.36	1.11	1.01	1.00	1.20	1.01	1.14	2.40	1.02
95	11.0	2.59	1.42	1.13	8.92	1.35	5.46	131	1.18
100	14.2	45.3	24.1	8.00	3.1e4	11.8	9.9e3	1.4e5	13.6

Table 24.4 Fitting a fifth degree polynomial $f(x)$ with $\mathrm{P}_X = U(0,1)$ and $\sigma = 0.05$. Table gives distribution of approximation ratios achieved at training sample size $l = 30$, showing percentiles of approximation ratios achieved in 1000 repeated trials.

CVT made significant over-fitting errors from time to time. This is evidenced by the fact that in 1000 trials with a training sample of size 30 (Table 24.1) TRI produced a *maximum* approximation ratio of 2.18, whereas CVT produced a worst case approximation ratio of 643, and the penalization strategies SRM and GCV both produced worst case ratios of 1.6×10^7. The 95th percentiles were TRI 1.45, CVT 6.11, SRM 419, GCV 2.7×10^3. Similar results for TRI are obtained for larger labeled sample sizes, such as $l = 100$ and $l = 200$. [4] For a broader selection of results see [SS02].

The results showing TRI's robustness against over-fitting are encouraging but it is further possible to prove that TRI cannot produce an approximation ratio greater than 3 due to over-fitting. That is, we can bound TRI's approximation ratio under two simple assumptions. First, that TRI makes it to the best hypothesis h_m in the sequence. Second, that the empirical error of h_m is an underestimate—that is, $\hat{d}(\mathrm{P}_{Y|X}, h_m) \leq d(\mathrm{P}_{Y|X}, h_m)$. Note that this second assumption is likely to hold because hypotheses are chosen by explicitly minimizing $\hat{d}(\mathrm{P}_{Y|X}, h_m)$ rather than $d(\mathrm{P}_{Y|X}, h_m)$ (see Table 24.5). The proof for the following proposition can be found in [SS02].

Proposition 24.1 *Let h_m be the optimal hypothesis in the sequence $h_0, h_1, ...$ (that is, $h_m = \arg\min_{h_k} d(\mathrm{P}_{Y|X}, h_k)$) and let h_ℓ be the hypothesis selected by TRI. If (i) $m \leq \ell$ and (ii) $\hat{d}(\mathrm{P}_{Y|X}, h_m) \leq d(\mathrm{P}_{Y|X}, h_m)$ then:*

$$d(\mathrm{P}_{Y|X}, h_\ell) \quad \leq \quad 3d(\mathrm{P}_{Y|X}, h_m) \tag{24.6}$$

Note that in Proposition 24.1, as well as in Propositions 24.2 and 24.3 below, it is implicitly assumed that the true inter-hypothesis distances $d(h_m, h_\ell)$ are known. This, in principle, must be measured on the true marginal P_X. This assumption will be relaxed in Section 24.3.4 below.

Continuing with the experimental investigation, the basic flavor of the 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 [SS02] and "difficult" target functions like $\sin(1/x)$ (Table 24.2). Here the complexity penalization methods (SRM, GCV, etc.) can be forced into a regime of constant catastrophe, CVT noticeably degrades, and yet TRI retains performance similar to the levels shown in Table 24.1.

Of course, these results might be due to considering a pathological target function from the perspective of polynomial curve fitting. It is therefore important to consider other more natural targets that might be better suited to polynomial approximation. In fact, by repeating the previous experiments with a more benign target function, $f(x) = \sin^2(2\pi x)$, quite different results are obtained. Table 24.3 shows that procedure TRI does not fare as well in this case—obtaining a median approximation ratio of 3.51 (compared to 1.03 for SRM, and 1.16 for CVT). A closer inspection of TRI's behaviour reveals that the reason for

[4] Although one might suspect that the large failures could be due to measuring relative instead of absolute error, it turns out that all of these large relative errors also correspond to large absolute errors. This is verified in Section 24.4.1 below.

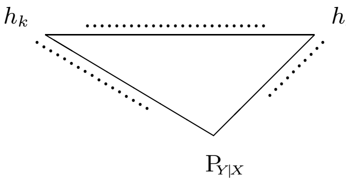


Figure 24.5 The real and estimated distances between successive hypotheses h_k and h_ℓ and the target $\mathrm{P}_{Y|X}$. Solid lines indicate real distances, dotted lines indicate empirical distance estimates.

this performance drop is that TRI systematically gets stuck at low even-degree polynomials (cf. Table 24.5). In fact, there is a simple geometric explanation for this. The even-degree polynomials (after degree 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 significant distance between successive polynomials and causes the triangle inequality test to fail between the even and odd degree fits, even though the larger even-degree polynomials give a good approximation. Therefore, although the metric-based TRI strategy is robust against over-fitting, it can be prone to systematic under-fitting in seemingly benign cases. Similar results were obtained for fitting a fifth degree target polynomial corrupted by the same level of Gaussian noise (Table 24.4). This problem demonstrates that the first assumption used in Proposition 24.1 above can be violated in natural situations (see Table 24.5). Consideration of this difficulty leads to the development of a reformulated procedure.

24.3.3 Strategy 2: Adjusted distance estimates

Assume for the sake of argument that $\tilde{d} = d$ (i.e., our estimate of inter-hypothesis distance, based on unlabeled data, is the true distance). The final idea explored for model selection is to observe that there would then be *two* metrics—the true metric d defined by the joint distribution P_{XY} and an empirical metric $\hat{d}$ determined by the labeled training sequence $\langle \mathbf{x}_1, y_1\rangle, \ldots, \langle \mathbf{x}_l, y_l\rangle$. Note that the previous model selection strategy TRI ignored the fact that one could measure the empirical distance between hypotheses $\hat{d}(h_k, h_\ell)$ on the *labeled* training data, as well as estimate their "true" distance $d(h_k, h_\ell)$ on the unlabeled data. However, the fact that one can measure both inter-hypothesis distances actually gives an *observable* relationship between $\hat{d}$ and d in the local vicinity. This observation is now exploited in an attempt to derive an improved model selection procedure.

Given the two metrics d and $\hat{d}$, consider the triangle formed by two hypotheses h_k and h_ℓ and the target conditional $\mathrm{P}_{Y|X}$ (Figure 24.5). Notice that there are six distances involved—three real and three estimated—of which the true distances to $\mathrm{P}_{Y|X}$ are the only two of importance, and yet these are the only two that are not available. However, the observed relationship between d and $\hat{d}$ can be exploited to adjust the empirical training error estimate $\hat{d}(\mathrm{P}_{Y|X}, h_\ell)$. In fact, one could first consider the simplest possible adjustment based on the naive assumption that the observed relationship of the metrics $\hat{d}$ and d between h_k and h_ℓ also holds between h_ℓ and $\mathrm{P}_{Y|X}$. Note that if this were actually the case, a better estimate of $d(\mathrm{P}_{Y|X}, h_\ell)$ could be obtained by simply re-scaling the training distance $\hat{d}(\mathrm{P}_{Y|X}, h_\ell)$ according to the observed ratio $\tilde{d}(h_k, h_\ell)/\hat{d}(h_k, h_\ell)$. Since $\hat{d}$ is expected to be

Procedure ADJ

- Given hypothesis sequence $h_0, h_1, ...$
- For each hypothesis h_ℓ in the sequence
  – multiply its estimated distance to the target $\hat{d}(\mathrm{P}_{Y|X}, h_\ell)$ by the worst ratio of unlabeled and labeled distance to some predecessor h_k to obtain an adjusted distance estimate:
  $$\check{d}(\mathrm{P}_{Y|X}, h_\ell) \;\triangleq\; \hat{d}(\mathrm{P}_{Y|X}, h_\ell)\frac{\tilde{d}(h_k, h_\ell)}{\hat{d}(h_k, h_\ell)}.$$
- Choose the hypothesis h_n with the smallest adjusted distance $\check{d}(h_n, \mathrm{P}_{Y|X})$.

Figure 24.6 Adjusted-distance-estimate model selection procedure.

an underestimate in general, because we assume the h_k are chosen by minimizing $\hat{d}$, this ratio should be larger than one. In fact, adopting this as a simple heuristic yields another model selection procedure, ADJ, which is also surprisingly effective (Figure 24.6). This simple procedure overcomes some of the under-fitting problems associated with TRI and yet retains much of TRI's robustness against over-fitting.

Although at first glance this procedure might seem to be ad hoc, it turns out that one can prove an over-fitting bound for ADJ that is analogous to that established for TRI. In particular, if one assumes that ADJ makes it to the best hypothesis h_m in the sequence, and the adjusted error estimate $\check{d}(\mathrm{P}_{Y|X}, h_m)$ is an underestimate, then ADJ cannot over-fit by a factor much greater than three. Again, the formal proposition is stated, but refer to [SS02] for a proof.

Proposition 24.2 *Let h_m be the optimal hypothesis in the sequence $h_0, h_1, \ldots$ and let h_ℓ be the hypothesis selected by ADJ. If (i) $m \leq \ell$ and (ii) $\check{d}(\mathrm{P}_{Y|X}, h_m) \leq d(\mathrm{P}_{Y|X}, h_m)$ then*

$$d(\mathrm{P}_{Y|X}, h_\ell) \quad \leq \quad \left(2 + \frac{\hat{d}(\mathrm{P}_{Y|X}, h_m)}{\hat{d}(\mathrm{P}_{Y|X}, h_\ell)}\right) d(\mathrm{P}_{Y|X}, h_m) \tag{24.7}$$

In this respect, not only does ADJ exhibit robustness against over-fitting, it also has a (weak) theoretical guarantee against under-fitting. That is, with the assumptions that the empirical distance estimates are underestimates and that the adjusted distance estimates strictly increase the empirical distance estimates, then if the true error of a successor hypothesis h_m improves the true error of all of its predecessors h_ℓ by a significant factor, h_m will be selected in lieu of its predecessors. See [SS02] for a proof of this proposition.

Proposition 24.3 *Consider a hypothesis h_m, and assume that (i) $\hat{d}(\mathrm{P}_{Y|X}, h_\ell) \leq d(\mathrm{P}_{Y|X}, h_\ell)$ for all $0 \leq \ell \leq m$, and (ii) $\hat{d}(\mathrm{P}_{Y|X}, h_\ell) \leq \check{d}(\mathrm{P}_{Y|X}, h_\ell)$ for all $0 \leq \ell < m$. Then if:*

$$d(\mathrm{P}_{Y|X}, h_m) \quad < \quad \frac{1}{3}\frac{\hat{d}(\mathrm{P}_{Y|X}, h_\ell)^2}{d(\mathrm{P}_{Y|X}, h_\ell)} \tag{24.8}$$

for all $0 \leq \ell < m$ (that is, $d(\mathrm{P}_{Y|X}, h_m)$ is sufficiently small) it follows that $\check{d}(\mathrm{P}_{Y|X}, h_m) < \check{d}(\mathrm{P}_{Y|X}, h_\ell)$ for all $0 \leq \ell < m$, and therefore ADJ will not choose any predecessor of h_m.

Therefore, although ADJ might not have originally appeared to be well motivated, it possesses worst case bounds against over-fitting and under-fitting that are different from those

	step$(x \geq 0.5)$ (Table 24.1)	$\sin(1/x)$ (Table 24.2)	$\sin^2(2\pi x)$ (Table 24.3)	poly$^5(x)$ (Table 24.4)
Proposition 24.1(i) holds	73	80	10	4
Proposition 24.1(ii) holds	87	86	99	98
Proposition 24.1 holds	61	66	9	4
Proposition 24.2(i) holds	27	32	28	67
Proposition 24.2(ii) holds	22	26	14	24
Proposition 24.2 holds	15	17	12	21

Table 24.5 Strengths of the assumptions used in Propositions 24.1 and 24.2. Table shows frequency (in percent) that the assumptions hold over 1000 repetitions of the experiments conducted in Tables 24.1, 24.2, 24.3 and 24.4 (at sample size $l = 20$).

that have been established for conventional methods. However, these bounds remain somewhat weak. Table 24.5 shows empirical results on the frequency with which the underlying assumptions hold on experimental data, demonstrating that both ADJ and TRI systematically under-fit in the experiments. That is, even though assumption (ii) of Proposition 24.1 is almost always satisfied (as expected), assumption (ii) of Proposition 24.2 is only true one quarter of the time. Therefore, Propositions 24.1 and 24.2 can only provide a loose characterization of the quality of these methods. However, both metric-based procedures remain robust against over-fitting.

To demonstrate that ADJ is indeed effective, the previous experiments were repeated with ADJ as a new competitor. The 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 domain distributions P_X. Tables 24.1–24.4 show the previous data along with the performance characteristics of ADJ. In particular, Tables 24.3–24.5 show that ADJ avoids the extreme under-fitting problems that hamper TRI; it appears to responsively select high order approximations when this is supported by the data. Moreover, Tables 24.1–24.2 show that ADJ is still extremely robust against over-fitting, even in situations where the standard approaches make catastrophic errors. Overall, this is the best model selection strategy observed for these polynomial regression tasks, even though it possesses a weaker guarantee against over-fitting than TRI [SS02].

Note that both proposed model selection procedures add little computational overhead to traditional methods, since computing inter-hypothesis distances involves making only a single pass down the reference list of unlabeled examples. This is an advantage over standard hold-out techniques like CVT which repeatedly call the hypothesis generating mechanism to generate pseudo-hypotheses—which can sometimes be expensive.

Finally, note that ADJ possesses a subtle limitation: the multiplicative re-scaling it employs cannot penalize hypotheses that have zero training error (hence the limiting of the degree of the polynomials to $l - 2$ in the above experiments to avoid null training errors). However, despite this shortcoming the ADJ procedure turns out to perform very well in experiments and most often outperforms the more straightforward TRI strategy.

	percentiles of approximation ratios				
$l = 30$	25	50	75	95	100
TRI $(u = 500)$	1.00	1.07	1.19	1.48	2.21
TRI $(u = 200)$	1.00	1.08	1.19	1.45	2.18
TRI $(u = 100)$	1.00	1.08	1.19	1.45	2.49
TRI $(u = \ \ 50)$	1.01	1.08	1.19	1.65	7.26
TRI $(u = \ \ 25)$	1.01	1.10	1.27	2.74	64.6
ADJ $(u = 500)$	1.06	1.14	1.26	1.51	1.99
ADJ $(u = 200)$	1.06	1.14	1.25	1.51	2.10
ADJ $(u = 100)$	1.07	1.16	1.31	1.67	2.21
ADJ $(u = \ \ 50)$	1.07	1.17	1.29	1.58	3.19
ADJ $(u = \ \ 25)$	1.09	1.22	1.40	1.85	8.68

Table 24.6 Fitting $f(x) = \text{step}(x \geq 0.5)$ with $\mathrm{P}_X = U(0, 1)$ and $\sigma = 0.05$ (as in Table 24.1). This table gives distribution of approximation ratios achieved with $l = 30$ labeled training examples and $u = 500$, $u = 200$, $u = 100$, $u = 50$, $u = 25$ unlabeled examples, showing percentiles of approximation ratios achieved after 1000 repeated trials. The experimental set up of Table 24.1 is repeated, except that a smaller number of unlabeled examples are used.

24.3.4 Robustness to unlabeled data

Before moving on to regularization, a comment on the robustness of these model selection techniques to limited amounts of auxiliary unlabeled data. In principle, one can always argue that the preceding empirical results are not useful because the metric-based strategies TRI and ADJ might require significant amounts of unlabeled data to perform well in practice. However, the 200 unlabeled examples used in the previous experiments does not seem that onerous. In fact, the previous theoretical results (Propositions 24.1–24.3) assumed knowledge of the true marginal P_X. To explore the issue of robustness to limited amounts of unlabeled data, the previous experiments were repeated but TRI and ADJ were only given a small auxiliary sample of unlabeled data to estimate inter-hypothesis distances. In this experiment it was found that these strategies were actually quite robust to using approximate distances. Table 24.6 shows that small numbers of unlabeled examples were still sufficient for TRI and ADJ to perform nearly as well as before. Moreover, Table 24.6 shows that these techniques only seem to significantly degrade with fewer unlabeled than labeled training examples. This robustness was observed across the range of problems considered.

Although the empirical results in this section are anecdotal, the paper [SUF97] pursues a more systematic investigation of the robustness of these procedures and reaches similar conclusions (also based on artificial data). Recently, Bengio and Chapados have also found that using a density estimate for P_X based only on labeled data allows one to dispense with unlabeled data and, surprisingly, still achieve beneficial results [BC03]. Rather than present a detailed investigation of these model selection strategies in more serious case studies, the focus now changes to a further improvement to the basic method.

24.4 Regularization

One difficulty when doing model selection is that the generalization behaviour depends on the specific decomposition of the base hypothesis class into subclasses. That is, different decompositions of H can lead to different outcomes. To avoid this issue, the previous ideas need to be extended to a more general training criterion that uses unlabeled data to decide how to penalize *individual* hypotheses in the global space H. The main contribution of this section is a simple, generic training objective that can be applied to a wide variety of supervised learning problems.

As before, assume a sizable collection of unlabeled data that can now be used to globally penalize complex hypotheses. Specifically, an alternative training criterion can be formulated that measures the behaviour of individual hypotheses on both the labeled and unlabeled data. The intuition behind this criterion is simple—instead of minimizing empirical training error alone, also seek hypotheses that behave *similarly* both on and off the labeled training data. This objective arises from the observation that a hypothesis which fits the training data well but behaves erratically off the labeled training set is not likely to generalize to unseen examples. To detect such behaviour one can measure the distances of a hypothesis from a fixed simple "origin" function ϕ on both data sets. If a hypothesis is behaving erratically off the labeled training set then it is likely that these two distances will disagree. This effect is demonstrated in Figure 24.7 for two large-degree polynomials that both fit the labeled training data well but differ dramatically in their true error and their differences between distances, both on and off training set, to the origin function. Trivial origin functions are used throughout this section—such as the zero function, $\phi = 0$, or the constant function at the mean of the y labels, $\phi = \bar{y}$. In practice, these work quite well.

To formulate a concrete training objective first requires the following tentative measures:

- empirical training error plus an additive penalty

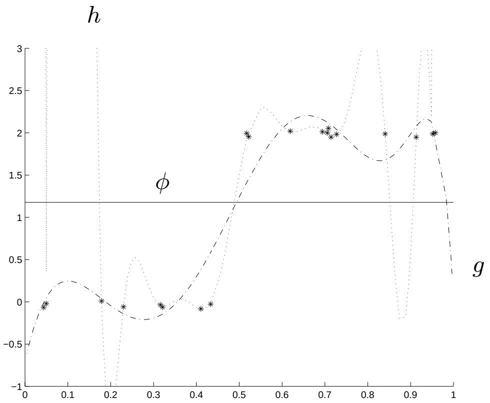
$$\hat{d}(h, \mathrm{P}_{Y|X}) + \tilde{d}(\phi, h) - \hat{d}(\phi, h) \tag{24.9}$$

- empirical error times a multiplicative penalty

$$\hat{d}(h, \mathrm{P}_{Y|X}) \times \frac{\tilde{d}(\phi, h)}{\hat{d}(\phi, h)} \tag{24.10}$$

In each case, the behaviour of a candidate hypothesis h is compared to the fixed origin ϕ. Thus, both cases will 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 distance to the origin on unlabeled data. The regularization effect of these criteria is illustrated in Figure 24.7. Somewhat surprisingly, the *multiplicative* objective (Equation 24.10) generally performs much better than the *additive* objective (Equation 24.9), as it more harshly penalizes discrepancies between on and off training set behaviour. Consequently, this is the form adopted from now on.

Although these training criteria might appear *ad hoc*, they are not entirely unprincipled. One useful property they have is that if the origin function ϕ happens to be equal to the tar-



$$\begin{aligned} \hat{d}(h, \mathrm{P}_{Y|X}) &= 0.004 \\ d(h, \mathrm{P}_{Y|X}) &= 193.1 \\ \hat{d}(g, \mathrm{P}_{Y|X}) &= 0.101 \\ d(g, \mathrm{P}_{Y|X}) &= 0.543 \end{aligned}$$

$$\begin{aligned} \hat{d}(\phi, h) &= 1.014 \\ \tilde{d}(\phi, h) &= 192.4 \\ \hat{d}(g, \phi) &= 1.010 \\ \tilde{d}(g, \phi) &= 0.928 \end{aligned}$$

Figure 24.7 Two nineteenth degree polynomials h and g that fit 20 given training points. Here h approximately minimizes $\hat{d}(h, \mathrm{P}_{Y|X})$, whereas g optimizes an alternative training criterion defined in (24.10). This plot demonstrates how the labeled training data estimate $\hat{d}(g, \mathrm{P}_{Y|X})$ for the smoother polynomial g is much closer to its true distance $d(g, \mathrm{P}_{Y|X})$. However, for both functions the proximity of the estimated errors $\hat{d}(\cdot, \mathrm{P}_{Y|X})$ to the true errors $d(\cdot, \mathrm{P}_{Y|X})$ appear to be reflected on the relative proximity of the estimated distances $\hat{d}(\cdot, \phi)$ to the unlabeled distances $\tilde{d}(\cdot, \phi)$ to the simple constant origin function ϕ.

get conditional $\mathrm{P}_{Y|X}$, then minimizing Equation 24.9 or Equation 24.10 becomes equivalent to minimizing the true prediction error $d(h, \mathrm{P}_{Y|X})$. However, it turns out that these training objectives have the inherent drawback that they subtly bias the final hypotheses towards the origin function ϕ. That is, both Equation 24.9 and Equation 24.10 allow minima that have "artificially" large origin distances on the labeled data, $\hat{d}(\phi, h)$, and simultaneously small distances on unlabeled data, $\tilde{d}(\phi, h)$. This is illustrated in Figure 24.7 for a hypothesis function g that minimizes Equation 24.10 but is clearly attracted to the origin, ϕ, at the right end of the domain (off of the labeled training data).

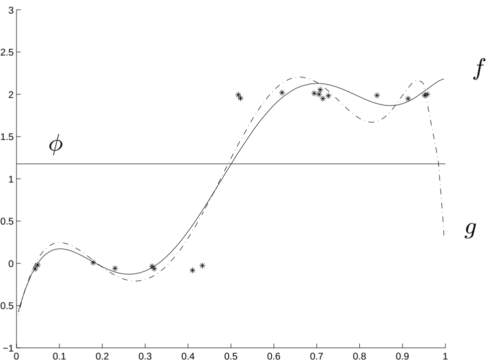
Nevertheless, there is an intuitive way to counter this difficulty. To avoid the bias towards ϕ, one can use *symmetric* forms of the previous criteria that also penalize hypotheses that are unnaturally *close* to the origin off of the labeled data. That is, one could consider a symmetric form of the additive penalty (Equation 24.9)

$$\hat{d}(h, \mathrm{P}_{Y|X}) + \left| \tilde{d}(\phi, h) - \hat{d}(\phi, h) \right| \tag{24.11}$$

as well as a symmetrized form of the multiplicative penalty (Equation 24.10)

$$\hat{d}(h, \mathrm{P}_{Y|X}) \times \max \left(\frac{\tilde{d}(\phi, h)}{\hat{d}(\phi, h)}, \frac{\hat{d}(\phi, h)}{\tilde{d}(\phi, h)} \right) \tag{24.12}$$

These penalties work in both directions: hypotheses that are much further from the origin on the training data than off are penalized, 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 behaviour indicate that the observed training error is likely to be an unrepresentative reflection of the true error of the hypothesis. The value of



$$\begin{aligned}
\hat{d}(g, \mathrm{P}_{Y|X}) &= 0.101\\
d(g, \mathrm{P}_{Y|X}) &= 0.543\\
\hat{d}(f, \mathrm{P}_{Y|X}) &= 0.098\\
d(f, \mathrm{P}_{Y|X}) &= 0.488\\
\\
\hat{d}(g, \phi) &= 1.010\\
\tilde{d}(g, \phi) &= 0.928\\
\hat{d}(f, \phi) &= 1.011\\
\tilde{d}(f, \phi) &= 1.023
\end{aligned}$$

Figure 24.8 A comparison of the asymmetric and symmetrized training objectives. Here g is the nineteenth degree polynomial which minimizes the original asymmetric criterion (24.10) on 20 data points, whereas f minimizes the symmetrized criterion (24.12). This plot shows how g is inappropriately drawn towards the origin ϕ near the right end of the interval, whereas f behaves neutrally with respect to ϕ.

this intuition is demonstrated in Figure 24.8, where the hypothesis f that minimizes the new symmetric criterion (Equation 24.12) is not drawn towards the origin inappropriately, and thereby achieves a smaller true prediction error than the hypothesis g that minimizes Equation 24.10. More technical justifications for this criterion are offered in [SS02].

The final outcome is a new regularization procedure that uses the training objective from Equation 24.12 to penalize hypotheses based on the given training data and the unlabeled data. In effect, the resulting procedure uses the unlabeled data to automatically set the level of regularization for a given problem. This procedure has an additional advantage—since the penalization factor in Equation 24.12 also depends on the specific labeled training set under consideration, the resulting procedure regularizes in a data-dependent fashion. That is, the procedure adapts the penalization to a 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, such an improvement can be achieved in realistic hypothesis classes on real data sets—as shown in the next section.

One drawback with the minimization objective in Equation 24.12 is that it is not convex and therefore local minima likely exist. Typically one has to devise reasonable initialization and restart procedures to effectively minimize such an objective. Here we simply started the optimizer from the best fit polynomial of each degree, or in the case of RBF regularization (below), we started from a single initialization point. Once initialized, a standard optimization routine (Matlab 5.3 “fminunc”) was used to determine coefficients that minimized Equations 24.11 and 24.12. Although the non-differentiability of Equation 24.12 creates difficulty for the optimizer, it does not prevent reasonable results from being achieved. Therefore, we did not find it necessary to smooth the objective with a softmax, although this is a reasonable idea. Another potential problem could arise if h gets close to the origin

ϕ. However, since simple origins were chosen that were never near $\mathrm{P}_{Y|X}$, h was not drawn near ϕ in these experiments and thus the resultant numerical instability did not arise.

24.4.1 Example: Polynomial regression

The first supervised learning task considered is the polynomial regression problem from Section 24.3.2. The regularizer introduced above (Equation 24.12) turns out to perform very well in such problems. In this case, our training objective can be expressed as choosing a hypothesis to minimize:

$$\sum_{i=1}^{l}(h(\mathbf{x}_i)-y_i)^2/l \quad \times \quad \max\left(\frac{\sum_{j=l+1}^{n}(h(\mathbf{x}_j)-\phi(\mathbf{x}_j))^2/u}{\sum_{i=1}^{l}(h(\mathbf{x}_i)-\phi(\mathbf{x}_i))^2/l}, \frac{\sum_{i=1}^{l}(h(\mathbf{x}_i)-\phi(\mathbf{x}_i))^2/l}{\sum_{j=l+1}^{n}(h(\mathbf{x}_j)-\phi(\mathbf{x}_j))^2/u}\right)$$

where $\{\langle \mathbf{x}_i, y_i\rangle\}_{i=1}^{l}$ is the set of labeled training data, $\{\langle \mathbf{x}_j\rangle\}_{j=l+1}^{n}$ is the set of unlabeled examples, and ϕ is a fixed origin function (usually set to the constant function at the mean of the y labels). Note again that this training objective seeks hypotheses that fit the labeled training data well while simultaneously behaving similarly on labeled and unlabeled data.

To test the basic effectiveness of this approach, the experiments of Section 24.3.2 were repeated. The first class of methods compared against were the same *model selection* methods considered before: 10-fold cross validation CVT, structural risk minimization SRM [CMV97], RIC [FG94]; SMS [Shi81], GCV [CW79], BIC [Sch78], AIC [Aka74], CP [Mal73], FPE [Aka70], and the metric based model selection strategy, ADJ, introduced in Section 24.3.3. However, since none of the classical model selection methods performed competitively in these experiments, they are not reported here (see [SS02] for more complete results). Instead, for comparison, results are reported for the optimal model selector, OPT*, which makes an oracle choice of the best available hypothesis in any given model selection sequence based on the test data. In these experiments, the model selection methods considered polynomials of degree 0 to $l-2$.[5]

The second class of methods compared against were *regularization* methods that consider polynomials of maximum degree $l-2$ but penalize individual polynomials based on the size of their coefficients or their smoothness properties. The specific methods 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 [CM98], and Bayesian *maximum a posteriori* inference with zero-mean Gaussian priors on polynomial coefficients a_k with diagonal covariance matrix λI [Mac92]. Both of these methods require a regularization parameter λ to be set by hand. These methods are referred to as REG and MAP respectively.

To test the ability of the new regularization technique to automatically set the regular-

[5] Note that the degree is restricted to be less than $l-1$ to prevent the maximum degree polynomials from achieving zero training error which, as discussed in Section 24.3, destroys the regularization effect of the multiplicative penalty.

		mean	median	stdev
ADA (24.12)	$\phi =$ mean y	0.391	0.366	0.113
asymmetric (24.10)		0.403	0.378	0.111
REG	$\lambda = 1.0$	0.483	0.468	0.048
REG*		0.371	0.355	0.049
model sel	OPT*	0.387	0.374	0.076
	ADJ	0.458	0.466	0.112

Table 24.7 Fitting $f(x) = \text{step}(x \geq 0.5)$ with $\mathrm{P}_X = U(0,1)$ and $\sigma = 0.05$. Absolute test errors (true distances) achieved. Results of 1000 repeated trials. This repeats the conditions of Table 24.1.

		mean	median	stdev
ADA (24.12)	$\phi =$ mean y	0.444	0.425	0.085
asymmetric (24.10)		0.466	0.439	0.102
REG	$\lambda = 1.0$	0.484	0.473	0.040
REG*		0.429	0.424	0.041
model sel	OPT*	0.433	0.427	0.049
	ADJ	0.712	0.504	0.752

Table 24.8 Fitting $f(x) = \sin(1/x)$ with $\mathrm{P}_X = U(0,1)$ and $\sigma = 0.05$. Absolute test errors (true distances) achieved. Results of 1000 repeated trials. This repeats the conditions of Table 24.2.

		mean	median	stdev
ADA (24.12)	$\phi =$ mean y	0.107	0.081	0.066
asymmetric (24.10)		0.111	0.087	0.060
REG	$\lambda = 5.0$	0.353	0.341	0.040
REG*		0.140	0.092	0.099
model sel	OPT*	0.122	0.085	0.086
	ADJ	0.188	0.114	0.150

Table 24.9 Fitting $f(x) = \sin^2(2\pi x)$ with $\mathrm{P}_X = U(0,1)$ and $\sigma = 0.05$. Absolute test errors (true distances) achieved. Results of 1000 repeated trials. This repeats the conditions of Table 24.3.

		mean	median	stdev
ADA (24.12)	$\phi =$ mean y	0.077	0.060	0.090
asymmetric (24.10)		0.110	0.074	0.088
REG	$\lambda = 10^{-1}$	0.454	0.337	0.508
REG*		0.147	0.082	0.121
model sel	OPT*	0.071	0.060	0.071
	ADJ	0.116	0.062	0.188

Table 24.10 Fitting a fifth degree polynomial with $\mathrm{P}_X = U(0,1)$ and $\sigma = 0.05$. Absolute test errors (true distances) achieved. Results of 1000 repeated trials. This repeats the conditions of Table 24.4.

ization level, a range of (fourteen) regularization parameters λ were tried for the fixed regularization methods REG and MAP, showing the single best value of λ obtained on the test data. For comparison purposes, the results of the oracle regularizer, REG*, is also reported. This oracle selects the best λ value for *each* training set based on examining the test data (MAP* gives similar results here [SS02]). The experiments were conducted by repeating the conditions of Section 24.3.2. Specifically, Table 24.7 repeats Table 24.1 (fitting a step function), Table 24.8 repeats Table 24.2 (fitting $\sin(1/x)$), Table 24.9 repeats Table 24.3 (fitting $\sin^2(2\pi x)$), and Table 24.10 repeats Table 24.4 (fitting a fifth degree polynomial). The regularization criterion based on minimizing Equation 24.12 is listed as ADA in our figures (for "adaptive" regularization). Additionally, the asymmetric version of ADA (24.10) was tested to verify the benefits of the symmetrized criterion (24.12).

The results are positive. The new adaptive regularization scheme ADA performed the best among all procedures in these experiments. Tables 24.7–24.10 show that it outperformed the fixed regularization strategy REG for the best fixed choice of regularization parameter (λ), even though the optimal choice varies across problems. This demonstrates that ADA is able to effectively tune its penalization behaviour 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 behaviour to a specific training set, not just a given problem. In fact, ADA is competitive with the oracle regularizer REG* in these experiments, and even sometimes outperformed the oracle model selection strategy OPT*. The results also show that the asymmetric version of ADA based on (24.10) is inferior to the symmetrized version in these experiments, confirming our prior expectations.

24.4.2 Example: Radial basis function regression

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

$$h(\mathbf{x}) \;=\; \sum_{i=1}^{k} w_i \; g\left(\frac{\|\mathbf{x} - c_i\|}{\sigma}\right) \qquad (24.13)$$

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

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 labels. The best fit weight vector can be obtained by solving for $\mathbf{w}$ in:

$$\begin{bmatrix} g\left(\frac{\|\mathbf{x}_1-\mathbf{x}_1\|}{\sigma}\right) & \cdots & g\left(\frac{\|\mathbf{x}_1-\mathbf{x}_l\|}{\sigma}\right) \\ \vdots & & \vdots \\ g\left(\frac{\|\mathbf{x}_l-\mathbf{x}_l\|}{\sigma}\right) & \cdots & g\left(\frac{\|\mathbf{x}_l-\mathbf{x}_l\|}{\sigma}\right) \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_l \end{bmatrix} = \begin{bmatrix} y_1 \\ \vdots \\ y_l \end{bmatrix}$$

The solution is guaranteed to exist and be unique for distinct training points and most natural basis functions, including the Gaussian basis used here [Bis95].

Although exactly fitting data with RBF networks is natural, it has the problem of generally over-fitting the training data in the process of replicating the y labels. Many approaches therefore 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. [CM98, Bis95]. The simplest regularization approach is to add a ridge penalty to the weight vector, and minimize

$$\sum_{i=1}^{l}(h(\mathbf{x}_i) - y_i)^2 + \lambda \sum_{i=1}^{l} w_i^2 \tag{24.14}$$

where h is given as in Equation 24.13 [CM98]. An alternative approach is to add a non-parametric penalty on curvature [PG90], 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 visualization and experimentation with the data set.

This section investigates 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 the basic idea is to use unlabeled data to make these choices automatically and adaptively. ADA (Equation 24.12) is compared to a large number of ridge regularization procedures, each corresponding to the penalty in Equation 24.14 with different fixed choices of σ and λ—thirty five in total. To apply ADA in this case a standard optimizer was run over the parameter space (σ, λ) while explicitly solving for the $\mathbf{w}$ vector that minimized Equation 24.14 for each choice of σ and λ (this involved solving a linear system [CM98, Bis95]). Thus, given σ, λ and $\mathbf{w}$ Equation (24.12) could be calculated and the result supplied to the optimizer as the objective to be minimized.

A number of regression problems from the StatLib and UCI machine learning repositories were investigated.[6] In the experiments, a given data set was randomly split into training (1/10), unlabeled (7/10), and test (2/10) sets. Each of the methods was then run on this split—this process being repeated 100 times for each data set to obtain results. Tables 24.11–24.14 show that ADA regularization was able to choose width and regularization parameters that achieved effective generalization performance across a range of data sets. The loss for ADA and REG* are given at the top of each table and the loss for each fixed parameter setting is given below. The best such setting is italicized. Furthermore, all set-

[6] The URLs are lib.stat.cmu.edu and www.ics.uci.edu/∼mlearn/MLRepository.html.

ADA (24.12) 0.0197 ± 0.004 | REG* 0.0329 ± 0.009

REG	λ=0.0	0.1	0.25	0.5	1.0
σ= 0.0005	0.0363	0.0447	0.0482	0.0515	0.0554
0.001	0.0353	0.0435	0.0475	0.0512	0.0554
0.0025	*0.0350*	0.0425	0.0473	0.0514	0.0555
0.005	0.0359	0.0423	0.0475	0.0516	0.0554
0.0075	0.0368	0.0424	0.0478	0.0517	0.0553

Table 24.11 RBF results showing mean test errors (distances) on the AAUP data set (1074 instances on 12 independent attributes). Results are averaged over 100 splits of the dataset.

ADA (24.12) 0.034 ± 0.0046 | REG* 0.049 ± 0.0063

REG	λ=0.0	0.1	0.25	0.5	1.0
σ= 4	0.4402	0.04954	0.04982	0.05008	0.05061
6	0.3765	0.04952	0.04979	0.05007	0.05063
8	0.3671	*0.04951*	0.04979	0.05007	0.05069
10	0.3474	0.04952	0.04979	0.05007	0.05073
12	0.3253	0.04953	0.04979	0.05008	0.05079

Table 24.12 RBF results showing mean test errors (distances) on the ABALONE data set (1000 instances on 8 independent attributes). Results are averaged over 100 splits of the dataset.

ADA (24.12) 0.131 ± 0.0171 | REG* 0.125 ± 0.0151

REG	λ=0.0	0.1	0.25	0.5	1.0
σ= 0.1	0.1658	**0.1299**	0.1325	0.1341	0.1354
0.5	0.1749	**0.1294**	0.1321	0.1337	0.1352
1	0.1792	***0.1294***	0.1321	0.1336	0.1353
2	0.1837	**0.1296**	0.1322	0.1337	0.1356
4	0.1883	**0.1299**	0.1323	0.1339	0.1362

Table 24.13 RBF results showing mean test errors (distances) on the BODYFAT data set (252 instances on 14 independent attributes). Results are averaged over 100 splits of the dataset.

ADA (24.12) 0.150 ± 0.0212 | REG* 0.151 ± 0.0197

REG	λ=0.0	0.1	0.25	0.5	1.0
σ= 0.075	0.1619	0.15785	0.1614	0.1645	0.1679
0.1	0.1624	0.15779	0.1614	0.1645	0.1679
0.15	0.1633	*0.15776*	0.1615	0.1646	0.1680
0.2	0.1642	0.15777	0.1615	0.1647	0.1682
0.25	0.1649	0.15780	0.1616	0.1648	0.1683

Table 24.14 RBF results showing mean test errors (distances) on the BOSTON-C data set (506 instances on 12 independent attributes). Results are averaged over 100 splits of the dataset.

tings that outperform ADA are shown in bold. Therefore, tables showing few bold entries indicate that ADA is outperforming most fixed regularizers.

On these datasets, ADA performs better than any fixed regularizer on every problem (except BODYFAT). This shows that the adaptive criterion is not only effective at choosing good regularization parameters for a given problem, but can choose them adaptively based on the specific sample of training data given, yielding improvements over fixed regularizers.

24.5 Classification

The regularization approach developed in this chapter can also be applied to classification problems. For classification, the label set Y is usually a small discrete set and prediction error is typically measured by the misclassification loss, *err*$(\hat{y}, y) = 1_{(\hat{y} \neq y)}$. With this loss function, distances are measured by the disagreement probability $d(f, g) = \mathrm{P}_X(f(\mathbf{x}) \neq g(\mathbf{x}))$ [BDIK95]. Using this metric, the generic regularization objective from Equation 24.12 can be directly applied to classification problems. As it turns out, a direct application of our approach to this case gives poor results [SS02]. An intuitive explanation for this weakness is that classification functions are essentially histogram-like (i.e., piecewise constant), and this tends to limit the ability of unlabeled data to detect erratic behaviour off the labeled training sample. A recent generalization analysis by Kääriäinen and Langford [Kää05, KL05] suggests that effective model selection strategies might be achieved by using tight generalization bounds derived from unlabeled data as a complexity penalizer. This idea has yet to be investigated in detail however. Rather than pursue modified techniques for classification here, we instead consider a straightforward regression-based approach for the remainder of this chapter.

A natural alternative to misclassification loss exists for the subset of classification methods that return a distribution over class labels instead of a single class label. With these methods, Kullback-Leibler (KL) divergence [CT91] can be used instead of distance metrics to compare hypothesis functions with the origin function ϕ.[7] With such a distance, penalized training objectives[8] can be derived similar to Equations 24.11 and 24.12, the terms of which are:

$$\tilde{d}(\phi \| h) = \frac{1}{u} \sum_{i=l+1}^{n} \phi(\mathbf{x}_i) \log \frac{\phi(\mathbf{x}_i)}{h(\mathbf{x}_i)} + (1 - \phi(\mathbf{x}_i)) \log \frac{1 - \phi(\mathbf{x}_i)}{1 - h(\mathbf{x}_i)} \tag{24.15}$$

$$\hat{d}(\phi \| h) = \frac{1}{l} \sum_{i=1}^{l} \phi(\mathbf{x}_i) \log \frac{\phi(\mathbf{x}_i)}{h(\mathbf{x}_i)} + (1 - \phi(\mathbf{x}_i)) \log \frac{1 - \phi(\mathbf{x}_i)}{1 - h(\mathbf{x}_i)} \tag{24.16}$$

[7] Note that KL divergence is not a proper distance metric but it is frequently used in such contexts.

[8] For the sake of simplicity, only binary classifi cation is considered.

	Set1	Set2	Set3	Set4	Set5	Set7
ADA	0.653	0.379	0.687	0.325	0.042	0.188
$\lambda = 0$	***0.570***	1.006	18.89	2.788	0.982	1.042
0.1	**0.502**	0.621	4.765	1.607	0.692	0.734
0.5	**0.537**	0.524	4.142	1.254	0.667	0.696
1.0	**0.568**	0.494	3.878	1.120	0.660	0.684
2.0	**0.606**	0.472	3.617	1.001	0.655	0.675
5.0	0.655	*0.459*	3.278	0.874	*0.653*	0.667
10.0	0.682	0.460	*3.026*	*0.804*	0.654	*0.664*

Table 24.15 Logistic regression (LR) results for six book data sets showing mean testing error (log-loss) for ADA and regularized LR with various settings of λ.

$$\hat{d}(\mathrm{P}_{Y|X}\|h) = \frac{1}{l}\sum_{i=1}^{l} -y_i \log h(\mathbf{x}_i) - (1-y_i)\log(1-h(\mathbf{x}_i)) \qquad (24.17)$$

Experiments were run on three classifiers that return class-membership probabilities. The ADA penalization strategy was tested on logistic regression (LR) [HTF01], kernel logistic regression (KLR) [HT90], and a neural network (NN) [HTF01]. Experiments were run on the two data sets used throughout this book and on a set of UCI data sets. The LR prediction function h is:

$$h(\mathbf{x}) = \frac{1}{1+e^{-\mathbf{w}^T\mathbf{x}}} \qquad (24.18)$$

The prediction functions for KLR and neural networks are closely related in the experiments presented here. KLR simply kernalizes Equation 24.18. For the neural network used here, the activation function in the first layer is $tanh()$ and the output layer uses the logistic function in Equation 24.18. The ADA penalized objectives for all three are therefore very similar.

In all cases, gradient descent was used to optimize the ADA objective. We compare against regularized versions of LR, using the penalty term $\lambda\mathbf{w}^T\mathbf{w},\ 0 \leq \lambda$. All experiments were repeated ten times, and the average log-loss test error reported. 10 labeled training points and 100 unlabeled points were used during training, and the remaining points were used for testing.

The results for LR on the book data sets (sets 1 through 5 and 7—set 6 was omitted because it is non-binary; sets 8 and 9 were omitted due to excessive size/dimensionality respectively) are shown in Table 24.15 for a variety of λ settings. The results show that ADA-penalized LR is competitive on Set1 and beats the best fixed regularizer on all other sets. Results on six UCI data sets (AUSTRALIAN, CRX, DIABETES, FLARE, GERMAN, and PIMA) are shown in Table 24.16. Again, results are competitive, coming close to the best fixed regularizer in most cases and surpassing it on two data sets.

Similar experiments were run on kernel logistic regression using a Gaussian kernel and a variety of settings for the standard deviation, σ. Results are shown in Table 24.17 for the

	AUST.	CRX	DIAB.	FLARE	GERM.	PIMA
ADA	0.697	0.716	0.703	0.541	0.697	0.683
$\lambda = 0$	1.240	1.176	1.282	1.741	0.710	1.442
0.1	0.927	0.797	0.785	0.833	0.715	0.881
0.5	0.814	0.707	0.733	0.618	0.715	0.773
1.0	0.773	0.689	0.716	0.572	0.713	0.739
2.0	0.742	0.679	0.703	0.546	0.710	0.715
5.0	0.715	*0.676*	0.694	0.533	0.703	0.697
10.0	*0.704*	0.678	*0.691*	*0.531*	*0.697*	*0.692*

Table 24.16 Logistic regression (LR) results for six UCI data sets showing mean testing error for ADA and regularized LR with various settings of λ.

book data sets [9] and in Table 24.18 for the UCI data. Like the earlier regression results, the best fixed parameter setting is italicized and all settings that outperform ADA are shown in bold.

On the book data, the results are excellent, beating the oracle regularizer on all but Set2 and coming very close even there. On the UCI data, the results are more mixed but still quite positive. While the oracle is not surpassed on any dataset, ADA is still better than many fixed regularizers.

Finally, we present results on three un-regularized neural networks, with three, five, and ten hidden units respectively. Results for the book data are shown in Table 24.19 and for the UCI data in Table 24.20. The results against un-regularized NN are striking, dramatically reducing the tendency to over-fit, even as the model complexity increases (performance on the PIMA data set with ten hidden nodes is the only notable anomaly to be found).

Overall, these results show considerable promise for the use of ADA with probabilistic classifiers but there clearly improvements still to be made. Adapting the technique to work with discrete classifiers also remains as a key challenge.

24.6 Conclusion

A new approach to the classical complexity-control problem has been introduced that is based on the intrinsic geometry of the function-learning task. This geometry is exploited in such a way as to be able to incorporate information from both labeled and unlabeled data in a semi-supervised learning task. Unlike the majority of such techniques, this approach requires no assumptions about the relationship between labeled and unlabeled data other than the key assumption that they are drawn from the same probability distribution.

These new techniques seem to outperform standard approaches in a wide range of regression problems and either outperform or are competitive with standard approaches in

[9] We presume the similar scores achieved by so many of the fixed regularizes on the book data are due to some regularity in that data.

	Set1			ADA 0.518	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	0.693	0.691	0.572	*0.569*	0.701
0.1	0.693	0.692	0.636	0.690	0.723
0.5	0.693	0.693	0.667	0.716	0.725
1.0	0.693	0.693	0.677	0.718	0.724
2.0	0.693	0.693	0.684	0.717	0.721
5.0	0.693	0.693	0.689	0.712	0.715
10.0	0.693	0.693	0.691	0.706	0.709

	Set2			ADA 0.456	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	0.693	0.693	0.691	0.478	0.480
0.1	0.693	0.693	0.692	***0.444***	0.477
0.5	0.693	0.693	0.693	0.481	0.498
1.0	0.693	0.693	0.693	0.503	0.504
2.0	0.693	0.693	0.693	0.531	0.511
5.0	0.693	0.693	0.693	0.578	0.526
10.0	0.693	0.693	0.693	0.615	0.549

	Set3			ADA 0.685	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
0.1	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
0.5	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
1.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
2.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
5.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
10.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*

	Set4			ADA 0.580	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	*0.693*	*0.693*	*0.693*	0.811	1.045
0.1	*0.693*	*0.693*	*0.693*	0.710	0.769
0.5	*0.693*	*0.693*	*0.693*	0.697	0.731
1.0	*0.693*	*0.693*	*0.693*	0.695	0.721
2.0	*0.693*	*0.693*	*0.693*	0.694	0.713
5.0	*0.693*	*0.693*	*0.693*	*0.693*	0.704
10.0	*0.693*	*0.693*	*0.693*	*0.693*	0.698

	Set5			ADA 0.513	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	*0.693*	*0.693*	*0.693*	*0.693*	0.754
0.1	*0.693*	*0.693*	*0.693*	*0.693*	0.701
0.5	*0.693*	*0.693*	*0.693*	*0.693*	0.695
1.0	*0.693*	*0.693*	*0.693*	*0.693*	0.694
2.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
5.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
10.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*

	Set7			ADA 0.514	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	*0.693*	*0.693*	*0.693*	*0.693*	0.736
0.1	*0.693*	*0.693*	*0.693*	*0.693*	0.697
0.5	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
1.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
2.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
5.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*
10.0	*0.693*	*0.693*	*0.693*	*0.693*	*0.693*

Table 24.17 Kernel logistic regression (KLR) results for six book data sets showing mean testing error for ADA and regularized KLR with various settings of λ and σ.

a range of classification problems, with only one comparatively weak instance (ADA regularized KLR). The primary source of this advantage is that the proposed metric-based strategies are able to detect dangerous situations and avoid making catastrophic over-fitting errors while still being responsive enough to adopt reasonably complex models when this is supported by the data. This is accomplished by attending to the real distances between hypotheses. Standard complexity-penalization strategies completely ignore this information. 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 [Sch94] and a universal improvement cannot be claimed for every complexity-control problem [Sch93], one should be able to exploit additional information about the task (i.e., knowledge of P_X) to obtain significant improvements across

	AUSTRALIAN			ADA 0.685	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	0.851	0.772	0.748	0.708	0.710
0.1	**0.670**	**0.681**	**0.682**	0.705	0.705
0.5	***0.653***	**0.671**	**0.682**	0.703	0.705
1.0	**0.654**	**0.671**	**0.683**	0.702	0.704
2.0	**0.658**	**0.673**	0.685	0.701	0.703
5.0	**0.667**	**0.674**	0.685	0.697	0.699
10.0	**0.675**	**0.677**	0.685	0.694	0.696

	CRX			ADA 1.111	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	1.141	1.153	**1.033**	**0.946**	**0.851**
0.1	**0.770**	**0.826**	**0.826**	**0.830**	**0.787**
0.5	**0.703**	**0.760**	**0.779**	**0.798**	**0.779**
1.0	**0.689**	**0.739**	**0.762**	**0.784**	**0.772**
2.0	**0.681**	**0.721**	**0.744**	**0.767**	**0.762**
5.0	***0.679***	**0.700**	**0.720**	**0.742**	**0.742**
10.0	**0.682**	**0.690**	**0.704**	**0.723**	**0.725**

	DIABETES			ADA 0.666	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	0.683	0.897	0.933	0.744	0.694
0.1	0.683	**0.638**	0.692	0.685	0.686
0.5	0.688	***0.619***	**0.658**	0.680	0.687
1.0	0.690	**0.623**	**0.649**	0.678	0.685
2.0	0.691	**0.633**	**0.645**	0.675	0.681
5.0	0.692	**0.652**	**0.645**	0.669	0.674
10.0	0.693	**0.666**	**0.652**	0.666	0.670

	FLARE			ADA 0.540	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	0.700	0.660	0.652	0.646	**0.473**
0.1	0.656	0.636	0.558	***0.465***	**0.474**
0.5	0.667	0.656	0.592	**0.468**	**0.481**
1.0	0.675	0.667	0.616	**0.474**	**0.483**
2.0	0.682	0.677	0.639	**0.482**	**0.485**
5.0	0.688	0.686	0.664	**0.500**	**0.494**
10.0	0.690	0.689	0.676	**0.526**	**0.511**

	GERMAN			ADA 0.804	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	0.968	1.480	1.574	0.888	**0.720**
0.1	**0.717**	0.814	0.845	**0.680**	**0.640**
0.5	**0.683**	**0.699**	**0.716**	**0.640**	**0.633**
1.0	**0.682**	**0.678**	**0.683**	**0.632**	**0.633**
2.0	**0.684**	**0.669**	**0.664**	***0.628***	**0.633**
5.0	**0.688**	**0.670**	**0.657**	***0.628***	**0.634**
10.0	**0.690**	**0.676**	**0.661**	**0.632**	**0.636**

	PIMA			ADA 0.680	
$\sigma =$	0.1	0.5	1	5	10
$\lambda = 0$	**0.678**	0.906	0.818	0.714	0.684
0.1	**0.679**	**0.646**	**0.679**	0.683	0.682
0.5	0.686	***0.636***	**0.670**	0.680	0.681
1.0	0.688	**0.641**	**0.666**	**0.679**	**0.679**
2.0	0.690	**0.648**	**0.663**	**0.677**	**0.678**
5.0	0.692	**0.661**	**0.661**	**0.673**	**0.674**
10.0	0.693	**0.672**	**0.664**	**0.671**	**0.671**

Table 24.18 Kernel logistic regression (KLR) results for six UCI data sets showing mean testing error for ADA and regularized KLR with various settings of λ and σ.

a wide range of problem types and conditions. The empirical results support this view. Furthermore, ADJ remains very competitive with newer model-selection techniques [BC03]. Additionally, ADJ has been independently extended along three lines [CVB02]: (i) producing excellent results on time-series data, (ii) using estimated densities in lieu of unlabeled data, and (iii) hybridizing ADJ with cross-validation.

An important direction for future research is to develop theoretical support for these strategies—in particular, a stronger theoretical justification of the regularization methods proposed in Section 24.4, an improved analysis of the model selection methods proposed in Section 24.3, and investigation of how to apply the technique in Section 24.5 to a more general set of classifiers . It remains open as to whether the proposed methods TRI, ADJ,

hidden=3	Set1	Set2	Set3	Set4	Set5	Set7
ADA	0.756	0.579	11.282	1.162	2.120	1.108
unreg NN	84.567	51.020	22.769	154.388	122.308	160.653
hidden=5	Set1	Set2	Set3	Set4	Set5	Set7
ADA	0.829	1.422	2.998	1.324	30.349	3.108
unreg NN	77.577	47.166	41.629	165.090	151.790	139.809
hidden=10	Set1	Set2	Set3	Set4	Set5	Set7
ADA	1.828	9.985	2.070	0.993	4.742	1.253
unreg NN	83.693	61.913	24.233	118.572	124.658	142.555

Table 24.19 NN results for the book data sets (except set 6) showing mean testing error for ADA and un-regularized NN with 3, 5 and 10 hidden nodes.

hidden=3	AUST.	CRX	DIAB.	FLARE	GERM.	PIMA
ADA	0.90	0.78	2.45	0.64	0.64	0.93
unreg NN	34.40	79.53	13.95	40.73	0.64	8.87
hidden=5	AUST.	CRX	DIAB.	FLARE	GERM.	PIMA
ADA	1.53	1.19	1.71	0.53	0.82	0.89
unreg NN	41.13	88.43	46.47	62.41	**0.73**	58.43
hidden=10	AUST.	CRX	DIAB.	FLARE	GERM.	PIMA
ADA	1.09	1.33	2.10	0.72	1.03	11.64
unreg NN	110.13	48.96	30.23	80.88	13.89	55.94

Table 24.20 NN results for six UCI data sets showing mean testing error for ADA and un-regularized NN with 3, 5 and 10 hidden nodes.

and ADA are in fact the best possible ways to exploit the hypothesis distances provided by P_X. A clear direction for future research is the investigation of alternative strategies that could potentially be more effective in this regard. For example, it remains future work to extend the multiplicative ADJ and ADA methods to cope with zero training errors. Additionally, more exploration of the effects of alternative origin functions (perhaps even ensembles of origin functions) is necessary. Finally, it would be interesting to adapt the approach to model combination methods, extending the ideas of [KV95] to other combination strategies, including boosting [FS97] and bagging [Bre96].

Acknowledgments

Research supported by the Alberta Ingenuity Centre for Machine Learning, NSERC, MITACS, and the Canada Research Chair programme.

References

Aka70. H. Akaike. Statistical predictor information. *Annals of the Institute of Statistical Mathematics*, 22:203–271, 1970.

Aka74. H. Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

BC03. Y. Bengio and N. Chapados. Extensions to metric based model selection. *J. Mach. Learn. Res.*, 3:1209–1227, 2003.

BDIK95. S. Ben-David, A. Itai, and E. Kushilevitz. Learning by distances. *Information and Computation*, 117(2):240–250, 1995.

Bis95. C. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995.

Bre96. L. Breiman. Bagging predictors. *Machine Learning*, 24:123–40, 1996.

CM98. V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods*. Wiley, New York, 1998.

CMV97. V. Cherkassky, F. Mulier, and V. Vapnik. Comparison of VC-method with classical methods for model selection. In *Proceedings World Congress on Neural Networks*, pages 957–962, 1997.

CT91. T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, New York, 1991.

CVB02. O. Chapelle, V. Vapnik, and Y. Bengio. Model selection for small sample regression. *Machine Learning*, 48(1-3):9–23, 2002.

CW79. P. Craven and G. Wahba. Smoothing noisy data with spline functions. *Numerische Mathematik*, 31:377–403, 1979.

Efr79. B. Efron. Computers and the theory of statistics: Thinking the unthinkable. *SIAM Review*, 21:460–480, 1979.

FG94. D. Foster and E. George. The risk inflation criterion for multiple regression. *Annals of Statistics*, 22:1947–1975, 1994.

FS97. Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.

GRV96. C. Galarza, E. Rietman, and V. Vapnik. Applications of model selection techniques to polynomial approximation. Preprint, 1996.

HT90. T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman and Hall, New York, 1990.

HTF01. T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer, 2001.

Kää05. M. Kääriäinen. Generalization error bounds using unlabeled data. In *Proceedings Annual Conference on Computational Learning Theory (COLT-05)*, 2005.

KL05. M. Kääriäinen and J. Langford. A comparison of tight generalization bounds. In *Proceedings International Conference on Machine Learning (ICML-05)*, 2005.

Koh95. R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-95)*, 1995.

KV95. A. Krogh and J. Vedelsby. Neural network ensembles, cross validation, and active learning. In *Advances in Neural Information Processing Systems 7*, pages 231–238, 1995.

Mac92. D. MacKay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.

Mal73. C. Mallows. Some comments on C_p. *Technometrics*, 15:661–676, 1973.

OS96. D. Opitz and J. Shavlik. Generating accurate and diverse members of a neural-network ensemble. In *Advances in Neural Information Processing Systems 8*, 1996.

PG90. T. Poggio and F. Girosi. Regularization algorithms for learning that are equivalent to multilayer networks. *Science*, 247:978–982, 1990.

Rip96. B. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996.

Ris86. J. Rissanen. Stochastic complexity and modeling. *Annals of Statistics*, 14:1080–1100, 1986.

Sch78. G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

Sch93. C. Schaffer. Overfitting avoidance as bias. *Machine Learning*, 10(2):153–178, 1993.

Sch94. C. Schaffer. A conservation law for generalization performance. In *Proceedings of International Conference on Machine Learning (ICML-94)*, pages 683–690, 1994.

Shi81. R. Shibata. An optimal selection of regression variables. *Biometrika*, 68:45–54, 1981.

SS02. D. Schuurmans and F. Southey. Metric-based methods for adaptive model selection and regularization. *Machine Learning*, 48(1-3):51–84, 2002. Special Issue on New Methods for Model Selection and Model Combination.

SUF97. D. Schuurmans, L. Ungar, and D. Foster. Characterizing the generalization performance of model selection strategies. In *Proceedings of International Conference on Machine Learning (ICML-97)*, pages 340–348, 1997.

Vap96. V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1996.

Vap98. V. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.