# An Experimental and Theoretical Comparison of Model Selection Methods*

Michael Kearns
AT&T Bell Laboratories
Murray Hill, New Jersey

Yishay Mansour
Tel Aviv University
Tel Aviv, Israel

Andrew Y. Ng
Carnegie Mellon University
Pittsburgh, Pennsylvania

Dana Ron
MIT
Cambridge, MA

**Abstract**

We investigate the problem of *model selection* in the setting of supervised learning of boolean functions from independent random examples. More precisely, we compare methods for finding a balance between the complexity of the hypothesis chosen and its observed error on a random training sample of limited size, when the goal is that of minimizing the resulting generalization error. We undertake a detailed comparison of three well-known model selection methods — a variation of Vapnik's *Guaranteed Risk Minimization* (GRM), an instance of Rissanen's *Minimum Description Length Principle* (MDL), and cross validation (CV). We introduce a general class of model selection methods (called *penalty-based* methods) that includes both GRM and MDL, and provide general methods for analyzing such rules. We provide both controlled experimental evidence and formal theorems to support the following conclusions:

- Even on simple model selection problems, the behavior of the methods examined can be both complex and incomparable. Furthermore, no amount of "tuning" of the rules investigated (such as introducing constant multipliers on the complexity penalty terms, or a distribution-specific "effective dimension") can eliminate this incomparability.
- It is possible to give rather general bounds on the generalization error, as a function of sample size, for penalty-based methods. The quality of such bounds depends in a precise way on the extent to which the method considered automatically limits the complexity of the hypothesis selected.
- For *any* model selection problem, the additional error of cross validation compared to *any* other method can be bounded above by the sum of two terms. The first term is large only if the learning curve of the underlying function classes experiences a "phase transition" between $(1-\gamma)m$ and $m$ examples (where $\gamma$ is the fraction saved for testing in CV). The second and competing term can be made arbitrarily small by increasing $\gamma$.
- The class of penalty-based methods is fundamentally handicapped in the sense that there exist two types of model selection problems for which every penalty-based method must incur large generalization error on at least one, while CV enjoys small generalization error on both.

Despite the inescapable incomparability of model selection methods under certain circumstances, we conclude with a discussion of our belief that the balance of the evidence provides specific reasons to prefer CV to other methods, unless one is in possession of detailed problem-specific information.

*This research was done while Y. Mansour, A. Ng and D. Ron were visiting AT&T Bell Laboratories.

# 1 Introduction

In the model selection problem (sometimes also known as complexity regularization), we must balance the complexity of a statistical model with its goodness of fit to the training data. This problem arises repeatedly in statistical estimation, machine learning, and scientific inquiry in general. Instances of the model selection problem include choosing the best number of hidden nodes in a neural network, determining the right amount of pruning to be performed on a decision tree, and choosing the degree of a polynomial fit to a set of points. In each of these cases, the goal is not to minimize the error on the training data, but to minimize the resulting generalization error.

The model selection problem is coarsely prefigured by Occam's Razor: given two hypotheses that fit the data equally well, prefer the simpler one. Unfortunately, Occam's Razor does not explicitly address the more complex, more interesting and more common problem in which we have a simple model with poor fit to the data, and a complex model with good fit to the data. Such a problem arises when the data is corrupted by noise, or when the size of the data set is small relative to the complexity of the process generating the data. Here we require not a qualitative statement of a preference for simplicity, but a *quantitative prescription* — a formula or algorithm — specifying the relative merit of simplicity and goodness of fit.

Many model selection algorithms have been proposed in the literature of several different research communities, too many to productively survey here. Various types of analysis have been used to judge the performance of particular algorithms, including asymptotic consistency in the statistical sense [17, 16], asymptotic optimality under coding-theoretic measures [12], and more seldom, rates of convergence for the generalization error [1]. Perhaps surprisingly, despite the many proposed solutions for model selection and the diverse methods of analysis, direct comparisons between the different proposals (either experimental or theoretical) are rare.

The goal of this paper is to provide such a comparison, and more importantly, to describe the general conclusions to which it has led. Relying on evidence that is divided between controlled experimental results and related formal analysis, we compare three well-known model selection algorithms. We attempt to identify their relative and absolute strengths and weaknesses, and we provide some general methods for analyzing the behavior and performance of model selection algorithms. Our hope is that these results may aid the informed practitioner in making an educated choice of model selection algorithm (perhaps based in part on some known properties of the model selection problem being confronted).

### Outline of the Paper

In Section 2, we provide a formalization of the model selection problem. In this formalization, we isolate the problem of choosing the appropriate *complexity* for a hypothesis or model. We also introduce the specific model selection problem that will be the basis for our experimental results, and describe an initial experiment demonstrating that the problem is nontrivial. In Section 3, we introduce the three model selection algorithms we examine in the experiments: Vapnik's Guaranteed Risk Minimization (GRM) [17], an instantiation of Rissanen's Minimum Description Length Principle (MDL) [12], and Cross Validation (CV).

Section 4 describes our controlled experimental comparison of the three algorithms. Using artificially generated data from a known target function allows us to plot complete learning curves for the three algorithms over a wide range of sample sizes, and to directly compare the resulting generalization error to the hypothesis complexity selected by each algorithm. It also allows us to

investigate the effects of varying other natural parameters of the problem, such as the amount of noise in the data. These experiments support the following assertions: the behavior of the algorithms examined can be complex and incomparable, even on simple problems, and there are fundamental difficulties in identifying a "best" algorithm; there is a strong connection between hypothesis complexity and generalization error; and it may be impossible to uniformly improve the performance of the algorithms by slight modifications (such as introducing constant multipliers on the complexity penalty terms).

In Sections 5, 6 and 7 we turn our efforts to formal results providing explanation and support for the experimental findings. We begin in Section 5 by upper bounding the error of any model selection algorithm falling into a wide class (called *penalty-based* algorithms) that includes both GRM and MDL (but not cross validation). The form of this bound highlights the competing desires for powerful hypotheses and controlled complexity. In Section 6, we upper bound the additional error suffered by cross validation compared to any other model selection algorithm. This quality of this bound depends on the extent to which the function classes have learning curves obeying a classical power law. Finally, in Section 7, we give an impossibility result demonstrating a fundamental handicap suffered by the entire class of penalty-based algorithms that does not afflict cross validation. In Section 8, we weigh the evidence and find that it provides concrete arguments favoring the use of cross validation (or at least cause for caution in using any penalty-based algorithm).

## 2 Definitions

Throughout the paper we assume that a fixed boolean *target function* $f$ is used to label inputs drawn randomly according to a fixed distribution $D$. For any boolean function $h$, we define the *generalization error*

$$\epsilon(h) \;=\; \epsilon_{f,D}(h) \;\stackrel{\text{def}}{=}\; \mathbf{Pr}_{x\in D}[h(x) \neq f(x)]^1. \tag{1}$$

We use $S$ to denote the random variable $S = \langle x_1, b_1\rangle, \ldots, \langle x_m, b_m\rangle$, where $m$ is the *sample size*, each $x_i$ is drawn randomly and independently according to $D$, and $b_i = f(x_i) \oplus c_i$, where the noise bit $c_i \in \{0,1\}$ is 1 with probability $\eta$; we call $\eta \in [0, 1/2)$ the *noise rate*. In the case that $\eta \neq 0$, we will sometimes wish to discuss the generalization error of $h$ with respect to the noisy examples, so we define

$$\epsilon^\eta(h) \stackrel{\text{def}}{=} \mathbf{Pr}_{x\in D,c}[h(x) \neq f(x) \oplus c], \tag{2}$$

where $c$ is the noise bit. Note that $\epsilon(h)$ and $\epsilon^\eta(h)$ are related by the equality

$$\begin{aligned} \epsilon^\eta(h) &= (1-\eta)\epsilon(h) + \eta(1-\epsilon(h)) \\ &= (1-2\eta)\epsilon(h) + \eta. \end{aligned} \tag{3}$$

Thus, $\epsilon^\eta(h)$ is simply a "damped" version of $\epsilon(h)$, and both quantities are minimized by the same $h$. For this reason, we use the term *generalization error* informally to refer to either quantity, making the distinction only when it is important.

[1] Except in circumstances where confusion may result, for brevity we shall adopt the notational convention of leaving implicit the many dependencies of the various quantities we define. Thus, we suppress the obvious dependence of $\epsilon(h)$ on $f$ and $D$, the dependence of empirical quantities on the random sample $S$, and so on.

We assume a nested sequence of *hypothesis classes* (or *models*)[2] $F_1 \subseteq \cdots \subseteq F_d \subseteq \cdots$. The target function $f$ may or may not be contained in any of these classes, so we define

$$h_d \stackrel{\text{def}}{=} \text{argmin}_{h \in F_d}\{\epsilon(h)\} \quad \text{and} \quad \epsilon_{opt}(d) \stackrel{\text{def}}{=} \epsilon(h_d) \tag{4}$$

(similarly, $\epsilon^{\eta}_{opt}(d) \stackrel{\text{def}}{=} \epsilon^{\eta}(h_d)$). Thus, $h_d$ is the best approximation to $f$ (with respect to $D$) in the class $F_d$, and $\epsilon_{opt}(d)$ measures the quality of this approximation. Note that $\epsilon_{opt}(d)$ is a non-increasing function of $d$ since the hypothesis function classes are nested. Thus, larger values of $d$ can only improve the *potential* approximative power of the hypothesis class. Of course, the difficulty is to realize this potential on the basis of a small sample.

With this notation, the model selection problem can be stated informally: on the basis of a random sample $S$ of a fixed size $m$, the goal is to choose a hypothesis *complexity* $\tilde{d}$, and a *hypothesis* $\tilde{h} \in F_{\tilde{d}}$, such that the resulting generalization error $\epsilon(\tilde{h})$ is minimized. In many treatments of model selection, including ours, it is explicitly or implicitly assumed that the model selection algorithm has control only over the choice of the complexity $\tilde{d}$, but not over the choice of the final hypothesis $\tilde{h} \in F_{\tilde{d}}$. It is assumed that there is a fixed algorithm that chooses a set of *candidate* hypotheses, one from each hypothesis class. Given this set of candidate hypotheses, the model selection algorithm then chooses one of the candidates as the final hypothesis.

To make these ideas more precise, we define the *training error*

$$\hat{\epsilon}(h) \;=\; \hat{\epsilon}_S(h) \;\stackrel{\text{def}}{=}\; |\{\langle x_i, b_i\rangle \in S : h(x_i) \neq b_i\}|/m, \tag{5}$$

and the *version space*

$$VS(d) \;=\; VS_S(d) \;\stackrel{\text{def}}{=}\; \{h \in F_d : \hat{\epsilon}(h) = \min_{h' \in F_d}\{\hat{\epsilon}(h')\}\}. \tag{6}$$

Note that $VS(d) \subseteq F_d$ may contain more than one function in $F_d$ — several functions may minimize the training error. If we are lucky, we have in our possession a (possibly randomized) *learning algorithm* $L$ that takes as input any sample $S$ and any complexity value $d$, and outputs a member $\tilde{h}_d$ of $VS(d)$ (using some unspecified criterion to break ties if $|VS(d)| > 1$). More generally, it may be the case that finding *any* function in $VS(d)$ is intractable, and that $L$ is simply a heuristic (such as backpropagation or ID3) that does the best job it can at finding $\tilde{h}_d \in F_d$ with small training error on input $S$ and $d$. In this paper we will consider both specific problems for which there is an efficient algorithm $L$ for selecting a function from the version space, and the more abstract case in which $L$ may be arbitrary. In either case, we define

$$\tilde{h}_d \stackrel{\text{def}}{=} L(S, d) \quad \text{and} \quad \hat{\epsilon}(d) = \hat{\epsilon}_{L,S}(d) \stackrel{\text{def}}{=} \hat{\epsilon}(\tilde{h}_d). \tag{7}$$

Note that we expect $\hat{\epsilon}(d)$, like $\epsilon_{opt}(d)$, to be a non-increasing function of $d$ — by going to a larger complexity, we can only reduce our training error. Indeed, we may even expect there to be a sufficiently large value $d_{\text{MAX}}$ (determined by the sequence of function classes, the learning algorithm, the target function and distribution) such that $\hat{\epsilon}(d_{\text{MAX}}) = 0$ always.

We can now give a precise statement of the model selection problem. First of all, an *instance* of the model selection problem consists of a tuple $(\{F_d\}, f, D, L)$, where $\{F_d\}$ is the hypothesis function class sequence, $f$ is the target function, $D$ is the input distribution, and $L$ is the underlying learning

[2]Such a nested sequence is called a *structure* by Vapnik [17] and is sometimes, but not always, the setting in which model selection methods are examined.

algorithm. The *model selection problem* is then: Given the sample $S$, and the sequence of functions $\tilde{h}_1 = L(S,1), \ldots, \tilde{h}_d = L(S,d), \ldots$ determined by the learning algorithm $L$, select a complexity value $\tilde{d}$ such that $\tilde{h}_{\tilde{d}}$ minimizes the resulting generalization error. Thus, a model selection algorithm is given both the sample $S$ and the sequence of (increasingly complex) hypotheses derived by $L$ from $S$, and must choose one of these hypotheses. Notice that "special" model selection criteria that incorporate knowledge about the behavior of the learning algorithm $L$ may be appropriate in certain cases; however, we hold that *good general model selection algorithms should at least perform reasonably well in the case that $L$ is actually a training error minimization procedure.*

The current formalization suffices to motivate a key definition and a discussion of the fundamental issues in model selection. We define

$$\epsilon(d) \;=\; \epsilon_{L,S}(d) \;\stackrel{\text{def}}{=}\; \epsilon(\tilde{h}_d). \tag{8}$$

Thus, $\epsilon(d)$ is a random variable (determined by the random variable $S$) that gives the *true generalization error* of the function $\tilde{h}_d$ chosen by $L$ from the class $F_d$. Of course, $\epsilon(d)$ is not directly accessible to a model selection algorithm; it can only be estimated or guessed in various ways from the sample $S$. A simple but important observation is that no model selection algorithm can achieve generalization error less than $\min_d\{\epsilon(d)\}$. Thus the behavior of the function $\epsilon(d)$ — especially the location and value of its minimum — is in some sense the essential quantity of interest in model selection.

The prevailing folk wisdom in several research communities posits the following picture for the "typical" behavior of $\epsilon(d)$, at least in the optimistic case that the learning algorithm $L$ implements training error minimization. (In the ensuing discussion, if there is classification noise the quantities $\epsilon^{\eta}_{opt}$ and $\epsilon^{\eta}$ should be substituted for $\epsilon_{opt}$ and $\epsilon$). First, for small values of $d$ ($d << m$), $\epsilon(d)$ is large, due simply to the fact that $\epsilon_{opt}(d)$ is large for small $d$, and $\epsilon(d) \geq \epsilon_{opt}(d)$ always holds. At such small $d$, training errors will be close to generalization errors (that is, $\hat{\epsilon}(h) \approx \epsilon(h)$ for all $h \in F_d$ — also known as *uniform convergence*, or small "variance"[3]), and $VS(d)$ will contain only functions whose true generalization error is near the best possible in $F_d$. But this best generalization error is large, because we have poor approximation power for small $d$ (that is, we have a strong "bias"). For large values of $d$ (usually $d \approx m$), $\epsilon(d)$ is again large, but for a different reason. Here we expect that $\epsilon_{opt}(d)$ may actually be quite small (that is, we have a weak "bias", and $F_d$ contains a good approximation to the target function $f$). But because $F_d$ is so powerful, $VS(d)$ will contain many poor approximations as well (that is, $VS(d)$ contains functions $h$ with $\hat{\epsilon}(h) << \epsilon(h)$ — so uniform convergence does *not* hold in $F_d$, or we have large "variance")[4].

As a demonstration of the validity of this view, and as an introduction to a particular model selection problem that we will examine in our experiments, we call the reader's attention to Figure 1. In this model selection problem (which we shall refer to as the *intervals model selection problem*), the input domain is simply the real line segment $[0,1]$, and the hypothesis class $F_d$ is simply the class of all boolean functions over $[0,1]$ in which we allow at most $d$ alternations of label; thus $F_d$ is the class of all binary step functions with at most $d/2$ steps. For the experiments, the underlying learning algorithm $L$ that we have implemented performs training error minimization. This is a rare case where efficient minimization is possible; we have developed an algorithm based on dynamic programming that runs in nearly linear time, thus making experiments on large samples feasible.

[3] We put the terms "bias" and "variance" in quotes in this paragraph to distinguish our informal use of them from their related but more precise statistical counterparts.

[4] A common way of informally expressing this behavior is to say that for small $d$, the functions in $VS(d)$ "underfit" the sample $S$, meaning that $F_d$ is not sufficiently expressive to capture the underlying regularities of $f$ exposed by $S$, and for large $d$, the functions in $VS(d)$ "overfit" the sample $S$.

The sample $S$ was generated using the target function in $F_{100}$ that divides $[0,1]$ into 100 segments of equal width $1/100$ and alternating label. (Details of the algorithm and the experimental results of the paper are provided in the Appendix.) In Figure 1 we plot $\epsilon(d)$ (which we can calculate exactly, since we have chosen the target function) when $S$ consists of $m = 2000$ random examples (drawn from the uniform input distribution) corrupted by noise at the rate $\eta = 0.2$. For our current discussion it suffices to note that $\epsilon(d)$ does indeed experience a nontrivial minimum. Not surprisingly, this minimum occurs near (but not exactly at) the target complexity of 100.

In Figure 2, we instead plot the difference $\hat{\epsilon}(d) - \epsilon(d)$ for the same experiments. Notice that there is something tempting about the simplicity of this plot. More precisely, as a function of $d/m$ it appears that $\hat{\epsilon}(d) - \epsilon(d)$ has an initial regime (for $d << 100$, or for this $m$, $d/m < 100/2000 = 0.05$) with behavior that is approximately $\Theta(\sqrt{d/m})$, and a later regime (for $d/m >> 0.05$) in which the behavior is linear in $d/m$. Unfortunately, the behavior near the target compexity $d = 100$ does not admit easy characterization. Nevertheless, Figure 2 demonstrates why one might be tempted to posit a "penalty" for complexity that is a function of $d/m$, and to simply add this penalty to $\hat{\epsilon}(d)$ as a rough approximation to $\epsilon(d)$.

According to Figure 1 and conventional wisdom, the best choice of $\tilde{d}$ should be an intermediate value (that is, *not* $\tilde{d} \approx 0$ or $\tilde{d} \approx m$). But how should we choose $\tilde{d}$ when the most common empirical measure of generalization ability — the function $\hat{\epsilon}(d)$ — simply decreases with increasing $d$, and whose straightforward minimization will therefore always result in a large value of $d$ that causes overfitting? This is the central question raised by the model selection problem, and many answers have been proposed and analyzed. We review three of them in the following section.

We conclude this section with a list of the various error measures that were presented in the section, and which are used extensively throughout the paper.

- $\epsilon(h)$ denotes the generalization error of a hypothesis $h$ with respect to the target function $f$ and the distribution $D$. Namely, $\epsilon(h) \stackrel{\text{def}}{=} \mathbf{Pr}_{x \in D}[h(x) \neq f(x)]$. Similarly, for noise rate $\eta > 0$, $\epsilon^{\eta}(h) \stackrel{\text{def}}{=} \mathbf{Pr}_{x \in D, c}[h(x) \neq f(x) \oplus c]$, where $c$ is the noise bit which is 1 with probability $\eta$, and 0 with probability $1 - \eta$.

- $\hat{\epsilon}(h)$ is the training error of $h$ on sample $S$. Namely, $\hat{\epsilon}(h) = \hat{\epsilon}_S(h) \stackrel{\text{def}}{=} |\{\langle x_i, b_i \rangle \in S : h(x_i) \neq b_i\}|/m$, where $m$ is the size of $S$.

- $\epsilon_{opt}(d)$ is the minimum generalization error taken over all hypotheses in $F_d$. Namely, $\epsilon_{opt}(d) \stackrel{\text{def}}{=} \epsilon(h_d)$, where $h_d \stackrel{\text{def}}{=} \text{argmin}_{h \in F_d}\{\epsilon(h)\}$. Similarly, $\epsilon^{\eta}_{opt}(d) \stackrel{\text{def}}{=} \epsilon^{\eta}(h_d)$.

- $\epsilon(d)$ is the generalization error of the hypothesis chosen by the learning algorithm $L$, in hypothesis class $F_d$, given sample $S$, and $\hat{\epsilon}(d)$ is the training error of the chosen hypothesis. Namely, $\epsilon(d) = \epsilon_{L,S}(d) \stackrel{\text{def}}{=} \epsilon(\tilde{h}_d)$, and $\hat{\epsilon}(d) = \hat{\epsilon}_{L,S}(d) \stackrel{\text{def}}{=} \hat{\epsilon}(\tilde{h}_d)$, where $\tilde{h}_d = L(S, d)$. $\epsilon^{\eta}(d)$ is defined analogously.

# 3 Three Algorithms for Model Selection

The first two model selection algorithms we consider are members of a general class that we shall informally refer to as *penalty-based* algorithms (and shall formally define shortly). The common theme behind these algorithms is their attempt to construct an approximation to $\epsilon(d)$ solely on the basis of the training error $\hat{\epsilon}(d)$ and the complexity $d$, often by trying to "correct" $\hat{\epsilon}(d)$ by the amount that it underestimates $\epsilon(d)$ through the addition of a "complexity penalty" term.

In Vapnik's *Guaranteed Risk Minimization* (GRM) [17], $\tilde{d}$ is chosen according to the rule

$$\tilde{d} = \text{argmin}_d \left\{ \hat{\epsilon}(d) + (d/m) \left( 1 + \sqrt{1 + \hat{\epsilon}(d)m/d} \right) \right\} \tag{9}$$

where we have assumed that $d$ is the Vapnik-Chervonenkis dimension [17, 18] of the class $F_d$; this assumption holds in the intervals model selection problem. Vapnik's original GRM actually multiplies the second term inside the argmin$\{\cdot\}$ above by a logarithmic factor intended to guard against worst-case choices from $VS(d)$, and thus has the following form:

$$\tilde{d} = \text{argmin}_d \left\{ \hat{\epsilon}(d) + \frac{d\left(\frac{\ln 2m}{d} + 1\right)}{m} \left( 1 + \sqrt{1 + \frac{\hat{\epsilon}(d)m}{d\left(\frac{\ln 2m}{d} + 1\right)}} \right) \right\} \tag{10}$$

However, we have found that the logarithmic factor renders GRM uncompetitive on the ensuing experiments, and hence in our experiments we only consider the modified and quite competitive rule given in Equation (9) whose spirit is the same. The origin of this rule can be summarized roughly as follows (where for sake of simplicity we use the term "with high probability" to mean with probability $1 - \delta$ over the draw of $S$, at a cost of a factor of $\log(1/\delta)$ in the bounds): it has been shown [17] that with high probability for every $d$ and for every $h \in F_d$, $\sqrt{d/m}$ is an upper bound on $|\hat{\epsilon}(h) - \epsilon(h)|$ and hence $|\hat{\epsilon}(d) - \epsilon(d)| \leq \sqrt{d/m}$. In fact, the stronger uniform convergence property holds: $|\hat{\epsilon}(h) - \epsilon(h)| \leq \sqrt{d/m}$ for all $h \in F_d$; the analogous statement holds for $\hat{\epsilon}(h)$ and $\epsilon^\eta(h)$ in the $\eta \neq 0$ case. Thus, by simply adding $\sqrt{d/m}$ to $\hat{\epsilon}(d)$, we ensure that the resulting sum upper bounds $\epsilon(d)$, and if we are optimistic we might further hope that the sum is in fact a close approximation to $\epsilon(d)$, and that its minimization is therefore tantamount to the minimization of $\epsilon(d)$. The actual rule given in Equation (9) is slightly more complex than this, and reflects a refined bound on $|\hat{\epsilon}(d) - \epsilon(d)|$ that varies from $d/m$ for $\hat{\epsilon}(d)$ close to 0 to $\sqrt{d/m}$ otherwise.

The next algorithm we consider, the *Minimum Description Length Principle* (MDL) [10, 11, 12, 1, 9] has rather different origins than GRM. MDL is actually a broad class of algorithms with a common information-theoretic motivation, each algorithm determined by the choice of a specific *coding* scheme for both functions and their training errors. This two-part code is then used to describe the training sample $S$. The familiar MDL motivation regards each potential hypothesis function as a code for the *labels* in the sample $S$, assuming the code recipient has access only to the inputs in $S$: thus, the "best" hypothesis is the one minimizing the total code length for the labels in the given coding scheme (the number of bits needed to represent the hypothesis function, plus the number of bits needed to represent the labels given the hypothesis function). To illustrate the method, we give a coding scheme for the intervals model selection problem[5]. Let $h$ be a function with exactly $d$ alternations of label (thus, $h \in F_d$). To describe the behavior of $h$ on the sample $S = \{\langle x_i, b_i \rangle\}$, we can simply specify the $d$ inputs where $h$ switches value (that is, the indices $i$ such that $h(x_i) \neq h(x_{i+1})$) [6]. This takes $\log \binom{m}{d}$ bits; dividing by $m$ to normalize, we obtain $(1/m) \log \binom{m}{d} \approx \mathcal{H}(d/m)$ [3], where $\mathcal{H}(\cdot)$ is the binary entropy function (i.e. $\mathcal{H}(p) \stackrel{\text{def}}{=} -(p \log p + (1-p)\log(1-p))$ ). Now given $h$, the labels in $S$ can be described simply by coding the mistakes of $h$ (that is, those indices $i$ where $h(x_i) \neq f(x_i)$), at a normalized cost of

[5] Our goal here is simply to give one reasonable instantiation of MDL. Other coding schemes are obviously possible; however, several of our formal results will hold for essentially all MDL instantiations.

[6] Notice that in this encoding, we are actually using the sample inputs to describe $h$. It is not difficult to see that under the assumption that the inputs are uniformly distributed in $[0, 1]$, this can be replaced by discretizing $[0, 1]$ using a grid of resolution $1/p(m)$, for some polynomial $p(\cdot)$, and using the grid points to describe the switches of $h$.

$\mathcal{H}(\hat{\epsilon}(h))$. Technically, in the coding scheme just described we also need to specify the values of $d$ and $\hat{\epsilon}(h) \cdot m$, but the cost of these is negligible. Thus, the version of MDL that we shall examine for the intervals model selection problem dictates the following choice of $\tilde{d}$:

$$\tilde{d} = \text{argmin}_d\{\mathcal{H}(\hat{\epsilon}(d)) + \mathcal{H}(d/m)\}. \tag{11}$$

In the context of model selection, GRM and MDL can both be interpreted as attempts to model $\epsilon(d)$ by some function of $\hat{\epsilon}(d)$ and $d$. More formally, a model selection algorithm of the form

$$\tilde{d} = \text{argmin}_d\{G(\hat{\epsilon}(d), d/m)\} \tag{12}$$

shall be called a *penalty-based* algorithm[7]. Notice that an ideal penalty-based algorithm would obey $G(\hat{\epsilon}(d), d/m) \approx \epsilon(d)$ (or at least $G(\hat{\epsilon}(d), d/m)$ and $\epsilon(d)$ would be minimized by the same value of $d$).

The third model selection algorithm that we examine has a different spirit than the penalty-based algorithms. In *cross validation* (CV) [15, 16], rather than attempt to *reconstruct* $\epsilon(d)$ from $\hat{\epsilon}(d)$ and $d$, we instead settle for a "worse" $\epsilon(d)$ (in a sense made precise shortly) that we can *directly estimate*. More specifically, in CV we use only a fraction $(1-\gamma)$ of the examples in $S$ to obtain the hypothesis sequence $\tilde{h}_1 \in F_1, \ldots, \tilde{h}_d \in F_d, \ldots$ — that is, $\tilde{h}_d$ is now $L(S', d)$, where $S'$ consists of the first $(1-\gamma)m$ examples in $S$. Here $\gamma \in [0,1]$ is a parameter of the CV algorithm whose tuning we discuss briefly later. CV chooses $\tilde{d}$ according to the rule

$$\tilde{d} = \text{argmin}_d\{\hat{\epsilon}_{S''}(\tilde{h}_d)\} \tag{13}$$

where $\hat{\epsilon}_{S''}(\tilde{h}_d)$ is the error of $\tilde{h}_d$ on $S''$, the last $\gamma m$ examples of $S$ that were withheld in selecting $\tilde{h}_d$. Notice that for CV, we expect the quantity $\epsilon(d) = \epsilon(\tilde{h}_d)$ to be (perhaps considerably) larger than in the case of GRM and MDL, because now $\tilde{h}_d$ was chosen on the basis of only $(1-\gamma)m$ examples rather than all $m$ examples. For this reason we wish to introduce the more general notation $\epsilon^{\gamma}(d) \stackrel{\text{def}}{=} \epsilon(\tilde{h}_d)$ to indicate the fraction of the sample withheld from training. CV settles for $\epsilon^{\gamma}(d)$ instead of $\epsilon^0(d)$ in order to have an independent test set with which to directly estimate $\epsilon^{\gamma}(d)$.

In practice, it is typical to use various forms of *multi-fold cross validation*, in which many (either disjoint or overlapping) training set/test set splits are selected from the original sample, and the test set errors are averaged. The main advantage of multi-fold methods is that each sample point is used for training on some splits; the main disadvantage is that the test sets are no longer independent. While we expect that for many problems, this lack of independence does not introduce diminished performance, we are unable to prove our general theoretical results for multi-fold methods, and thus concentrate on the basic cross-validation method outlined above. For this reason it is fair to say that we err on the side of pessimism when evaluating the performance of CV-type algorithms throughout the investigation.

# 4 A Controlled Experimental Comparison

Our results begin with a comparison of the performance and properties of the three model selection algorithms in a carefully controlled experimental setting — namely, the intervals model selection

[7] With appropriately modified assumptions, all of the formal results in the paper hold for the more general form $G(\hat{\epsilon}(d), d, m)$, where we decouple the dependence on $d$ and $m$. However, the simpler coupled form will usually suffice for our purposes.

problem. Among the advantages of such controlled experiments, at least in comparison to empirical results on data of unknown origin, are our ability to exactly measure generalization error (since we know the target function and the distribution generating the data), and our ability to precisely study the effects of varying parameters of the data (such as noise rate, target function complexity, and sample size), on the performance of model selection algorithms. The experimental behavior we observe foreshadows a number of important themes that we shall revisit in our formal results.

We begin with Figure 3. To obtain this figure, a training sample was generated from the uniform input distribution and labeled according to an intervals function over $[0, 1]$ consisting of 100 intervals of alternating label and equal width[8]; the sample was corrupted with noise rate $\eta = 0.2$. In Figure 3, we have plotted the *true* generalization errors (measured with respect to the noise-free source of examples) $\epsilon_{\mathrm{GRM}}$, $\epsilon_{\mathrm{MDL}}$ and $\epsilon_{\mathrm{CV}}$ (using test fraction $\gamma = 0.1$ for CV) of the hypotheses selected from the sequence $\tilde{h}_1, \ldots, \tilde{h}_d, \ldots$ by each the three algorithms as a function of the sample size $m$, which ranged from 1 to 3000 examples. As described in Section 2, the hypotheses $\tilde{h}_d$ were obtained by minimizing the training error within each class $F_d$. Details of the code used to perform these experiments is given in the appendix.

Figure 3 demonstrates the subtlety involved in comparing the three algorithms: in particular, we see that *none of the three algorithms outperforms the others for all sample sizes.* Thus we can immediately dismiss the notion that one of the algorithms examined can be said to be optimal for this problem in any standard sense. Getting into the details, we see that there is an initial regime (for $m$ from 1 to slightly less than 1000) in which $\epsilon_{\mathrm{MDL}}$ is the lowest of the three errors, sometimes outperforming $\epsilon_{\mathrm{GRM}}$ by a considerable margin. Then there is a second regime (for $m$ about 1000 to about 2500) where an interesting reversal of relative performance occurs, since now $\epsilon_{\mathrm{GRM}}$ is the lowest error, considerably outperforming $\epsilon_{\mathrm{MDL}}$, which has temporarily leveled off. In both of these first two regimes, $\epsilon_{\mathrm{CV}}$ remains the intermediate performer. In the third and final regime, $\epsilon_{\mathrm{MDL}}$ decreases rapidly to match $\epsilon_{\mathrm{GRM}}$ and the slightly larger $\epsilon_{\mathrm{CV}}$, and the performance of all three algorithms remains quite similar for all larger sample sizes.

Insight into the causes of Figure 3 is given by Figure 4, where for the same runs used to obtain Figure 3, we instead plot the quantities $\tilde{d}_{\mathrm{GRM}}$, $\tilde{d}_{\mathrm{MDL}}$ and $\tilde{d}_{\mathrm{CV}}$, the value of $\tilde{d}$ chosen by GRM, MDL and CV respectively (thus, the "correct" value, in the sense of simply having the same number of intervals as the target function, is 100). Here we see that for small sample sizes, corresponding to the first regime discussed for Figure 3 above, $\tilde{d}_{\mathrm{GRM}}$ is slowly approaching 100 from below, reaching and remaining at the target value for about $m = 1500$. Although we have not shown it explicitly, GRM is incurring nonzero training error throughout the entire range of $m$. In comparison, for a long initial period (corresponding to the first two regimes of $m$), MDL is simply choosing the shortest hypothesis that incurs no training error (and thus encodes both "legitimate" intervals and noise), and consequently $\tilde{d}_{\mathrm{MDL}}$ grows in an uncontrolled fashion. It will be helpful to compute an approximate expression for $\tilde{d}_{\mathrm{MDL}}$ during this "overcoding" period. Assuming that the target function is $s$ equally spaced intervals, an approximate expression for the number of intervals required to achieve zero training error is

$$d_0 \stackrel{\mathrm{def}}{=} 2\eta(1-\eta)m + (1-2\eta)^2 s. \tag{14}$$

For the current experiment $s = 100$ and $\eta = 0.2$. Equation (14) can be explained as follows. Consider the event that a given pair of consecutive inputs in the sample have opposite labels. If the two points belong to the same interval of the target function, then this event occurs if and only if exactly one of them is labeled incorrectly, which happens with probability $2\eta(1-\eta)$. If the two

[8]Similar results hold for a randomly chosen target function.

points are on opposite sides of a target switch in the target function, then this event occurs either if both of them are labeled correctly or if both of them are labeled incorrectly, which happens with probability $\eta^2 + (1-\eta)^2$. Since the expected number of pairs of the first type is $m - s$, and the expected number of pairs of the second type is $s$, we obtain (ignoring dependencies between the different pairs) that the expected number of switch points in the sample is roughly

$$\begin{aligned} 2\eta(1-\eta)(m-s) + (\eta^2 + (1-\eta)^2)s &= 2\eta(1-\eta)m + (1 - 4\eta + 4\eta^2)s & (15) \\ &= 2\eta(1-\eta)m + (1-2\eta)^2 s \quad = d_0 \, . & (16) \end{aligned}$$

In the first regime of Figures 3 and 4, the overcoding behavior $\tilde{d}_{\mathrm{MDL}} \approx d_0$ of MDL is actually preferable, in terms of generalization error, to the initial "undercoding" behavior of GRM, as verified by Figure 3. Once $\tilde{d}_{\mathrm{GRM}}$ approaches 100, however, the overcoding of MDL is a relative liability, resulting in the second regime. Figure 4 clearly shows that the transition from the second to the third regime (where approximate parity is achieved) is the direct result of a dramatic correction to $\tilde{d}_{\mathrm{MDL}}$ from $d_0$ (defined in Equation (14)) to the target value of 100. Finally, $\tilde{d}_{\mathrm{CV}}$ makes a more rapid but noisier approach to 100 than $\tilde{d}_{\mathrm{GRM}}$, and in fact also overshoots 100, but much less dramatically than $\tilde{d}_{\mathrm{MDL}}$. This more rapid initial increase again results in superior generalization error compared to GRM for small $m$, but the inability of $\tilde{d}_{\mathrm{CV}}$ to settle at 100 results in slightly higher error for larger $m$.

In a moment, we shall further discuss the interesting behavior of $\tilde{d}_{\mathrm{GRM}}$ and $\tilde{d}_{\mathrm{MDL}}$, but first we call attention to Figures 5 to 12. These figures, which come in pairs, show experiments identical to that of Figures 3 and 4, but for the smaller noise rates $\eta = 0.0, 0.1$ and the larger noise rates $\eta = 0.3, 0.4$; these plots also have an increased sample size range, $m = 1 \ldots 6500$. (Thus, the scale of these figures is different from that of Figures 3 and 4.) Notice that as $\eta$ increases, the initial period of undercoding by GRM seems to increase slightly, but the initial period of overcoding by MDL increases tremendously, the result being that the first regime of generalization error covers approximately the same values of $m$ (about 1 to 1000), but the second regime covers a wider and wider range of $m$, until at $\eta = 0.4$, $\tilde{d}_{\mathrm{MDL}}$ has not corrected to 100 even at $m = 6500$ (further experiments revealed that $m = 15000$ is still not sufficient).

The behavior of the lengths $\tilde{d}_{\mathrm{GRM}}$ and $\tilde{d}_{\mathrm{MDL}}$ in Figure 4 can be traced to the form of the total penalty functions for the two methods. For instance, in Figures 13, 14, and 15, we plot the total MDL penalty $\mathcal{H}(\hat{\epsilon}(d)) + \mathcal{H}(d/m)$ as a function of complexity $d$ for the fixed sample sizes $m = 500, 2000$ and 4000 respectively, again using noise rate $\eta = 0.20$. At $m = 500$, we see that the rather dramatic total penalty curve has its global minimum at approximately $d = 200$, which as expected (we are in the MDL overcoding regime at this small sample size) is $d_0$, the point of consistency with the noisy sample. However, a small local minimum is already developing near the target value of $d = 100$. By $m = 2000$, this local minimum is quite pronounced, and beginning to compete with the global consistency minimum (which for this noise rate and sample size has now moved out to approximately $d_0 = 650$). At $m = 4000$, the former local minimum at $d = 100$ has become the global minimum.

The rapid transition of $\tilde{d}_{\mathrm{MDL}}$ that marks the start of the final regime of generalization error discussed above (approximate parity of the three methods) is thus explained by the switching of the global total penalty minimum from $d_0$ to $d = 100$. From the expression given in Equation (14) we can infer that this switching of the minimum is governed by a competition between the quantities $\mathcal{H}(2\eta(1-\eta) + (s/m)(1-2\eta)^2)$ and $\mathcal{H}(\eta) + \mathcal{H}(s/m)$. The first quantity is the expected value of the total penalty of MDL for the choice $d = d_0$ (where the hypothesis chosen is consistent with the data and no training error is incurred), while the second quantity is the total penalty of MDL

for the (correct) choice $d = s$. As an interesting digression, in Figures 16, 17 and 18, we plot the difference $\mathcal{H}\left(2\eta(1-\eta) + (s/m)(1-2\eta)^2\right) - \left(\mathcal{H}(\eta) + \mathcal{H}(s/m)\right)$ as a function of $\eta$ for $s/m = 0.01$ and $s/m = 0.04$. Note that if this function is negative, we predict that MDL will prefer $d = d_0$ (overcoding), and if it is positive, we predict that MDL will prefer $d = s$. For $s/m = 0.01$, we see that the function is positive for small noise rates and negative for larger noise rates. Thus, make the intuitively reasonable prediction that for this value of the ratio $s/m$, increasing the noise rate can only degrade the behavior, by forcing the reversal of the global minimum from $d = s$ to $d = d_0$. Curiously, however, the difference exhibits nonmonotonic behavior as a function of $s/m$. For the case $s/m = 0.04$, this nonmonotonicity has a subtle but dramatic effect, since it causes the difference to move from negative to positive at small $\eta$. Thus we predict that for very small values of $\eta$ (less than 0.015), by *increasing* the noise rate slightly (that is, by adding a small amount of additional classification noise), we can actually cause the global minimum to shift from $d = d_0$ to $d = s$, and consequently improve the resulting generalization error. These predictions are in fact confirmed by experiments we conducted.

In Figures 19, 20, and 21, we give plots of the total GRM penalty for the same three sample sizes and noise rate. Here the behavior is much more controlled — for each sample size, the total penalty has the same single-minimum bowl shape, with the minimum starting to the left of $d = 100$ (the minimum occurs at roughly $d = 40$ for $m = 500$), and gradually moving over $d = 100$ and sharpening for large $m$.

A natural question to pose after examining these experiments is the following: is there a penalty-based algorithm that enjoys the best properties of both GRM and MDL? By this we would mean an algorithm that approaches the "correct" $d$ value (whatever it may be for the problem in hand) more rapidly than GRM, but does so without suffering the long, uncontrolled "overcoding" period of MDL. An obvious candidate for such an algorithm is simply a modified version of GRM or MDL, in which we reason (for example) that perhaps the GRM penalty for complexity is too large for this problem (resulting in the initial reluctance to code), and we thus multiply the complexity penalty term in the GRM rule (the second term inside the argmin$\{\cdot\}$) in Equation (9) by a constant less than 1 (or analogously, multiply the MDL complexity penalty term by a constant greater than 1 to reduce overcoding). The results of an experiment on such a modified version of GRM are shown in Figures 22 and 23, where the original GRM performance is compared to a modified version in which the complexity penalty is multiplied by 0.5. Interestingly and perhaps unfortunately, we see that there is no free lunch: while the modified version does indeed code more rapidly and thus reduce the small $m$ generalization error, this comes at the cost of a subsequent overcoding regime with a corresponding degradation in generalization error (and in fact a considerably slower return to $d = 100$ than MDL under the same conditions)[9]. The reverse phenomenon (reluctance to code) is experienced for MDL with an increased complexity penalty multiplier, as demonstrated by Figures 24 and 25. This observation seems to echo recent results [13, 19] which essentially prove that no learning algorithm can perform well on all problems. However, while these results show that for any given learning algorithm *there exist* learning problems (typically in which the target function is chosen randomly from a large and complex space) on which the performance is poor, here we have given an *explicit* and very simple learning problem on which no simple variant of GRM and MDL can perform well for all sample sizes.

Let us summarize the key points demonstrated by these experiments. First, none of the three algorithms dominates the others for all sample sizes. Second, the two penalty-based algorithms seem

[9]Similar results are obtained in experiments in which every occurrence of $d$ in the GRM rule is replaced by an "effective dimension" $c_0 d$ for any constant $c_0 < 1$.

to have a bias either towards or against coding that is overcome by the inherent properties of the data asymptotically, but that can have a large effect on generalization error for small to moderate sample sizes. Third, this bias cannot be overcome simply by adjusting the relative weight of error and complexity penalties, without reversing the bias of the resulting rule and suffering increased generalization error for some range of $m$. Fourth, while CV is not the best of the algorithms for any value of $m$, it does manage to fairly closely track the best penalty-based algorithm for each value of $m$, and considerably beats both GRM and MDL in their regimes of weakness. We now turn our attention to our formal results, where each of these key points will be developed further.

# 5 A Bound on the Error for Penalty-Based Algorithms

We begin our formal results with a bound on the generalization error for penalty-based algorithms that enjoys three features. First, it is general: it applies to practically any penalty-based algorithm, and holds for any model selection problem (of course, there is a price to pay for such generality, as discussed below). Second, for certain algorithms and certain problems the bound can give rapid rates of convergence to small error. Third, the form of the bound is suggestive of some of the behavior seen in the experimental results. Our search for a bound of this type was inspired by work of Barron and Cover [1]. Barron and Cover give bounds of a similar form (which they call the *index of resolution*) on the generalization error of MDL in the context of density estimation.

For a given penalty-based algorithm, let $G$ be the function that determines the algorithm as defined in Equation (12). In Theorem 1 we give a bound on the generalization error of such an algorithm, where the only restriction made on the algorithm is that $G$ be continuous and increasing in both its arguments. The bound we give consists of two terms. The first term, denoted by $R_G(m)$, is a function of the sample size, $m$, and as $m \to \infty$ it approaches the minimum generalization error achievable in *any* of the classes $F_d$. This minimum value, by definition, is a lower bound on the generalization error achieved by any possible method. Since the bound we give applies to quite a wide range of model selection algorithms, we are not able to provide a general statement concerning the *rate* of convergence of $R_G(m)$ to the optimal error, and this rate strongly depends on the properties of $G$. The general form of $R_G(m)$ (as a function of $G$ as well as $m$) is described in the proof of Theorem 1. Following the proof we discuss what properties must $G$ have in order that $R_G(m)$ converge at a reasonable rate to the optimal error. We also give several examples of the application of the theorem in which the exact form of $R_G(m)$ and hence its convergence rate become explicit. The second term in the bound is a function of $m$ as well, and it decreases very rapidly as $m$ increases. However, it is also an increasing function of the complexity chosen by the penalty-based algorithm, and thus, similarly to the first term, is dependent on the properties of $G$. We return to discuss this bound following the formal theorem statement below. We state the bound for the special but natural case in which the underlying learning algorithm $L$ is training error minimization. Towards the end of this section we present a straightforward analogue for more general $L$ (Theorem 2). In addition, we give a generalization of Theorem 1 to the noisy case (Theorem 3). In both theorems the bound given on the generalization error has a very similar form to the bound given in Theorem 1.

**Theorem 1** *Let $(\{F_d\}, f, D, L)$ be an instance of the model selection problem in which $L$ performs training error minimization, and where $d$ is the VC dimension of $F_d$. Let $G : [0,1] \times \Re \to \Re$ be a function that is continuous and increasing in both its arguments, and let $\epsilon_G(m)$ denote the expected generalization error of the penalty-based model selection algorithm $\tilde{d} = \mathrm{argmin}_d\{G(\hat{\epsilon}(d), d/m)\}$ on*

*a training sample of size $m$. Then for any given $\delta > 0$, with probability at least $1 - \delta$*

$$\epsilon_G(m) \leq R_G(m) + \beta(\tilde{d}, m, \delta) \,, \tag{17}$$

*where $R_G(m)$ approaches $\min_d\{\epsilon_{opt}(d)\}$ as $m \to \infty$, and where $\beta(\cdot,\cdot,\cdot)$ is defined as follows: for $d < m$,*

$$\beta(d, m, \delta) \stackrel{\text{def}}{=} 2\sqrt{\frac{d\ \ln\frac{2em}{d} + \ln\frac{9m}{\delta}}{m}} \tag{18}$$

*and for $d \geq m$, $\beta(d, m, \delta) \stackrel{\text{def}}{=} 1$.*

Before proving Theorem 1, let us further discuss the form of the bound given in the theorem. The first term, $R_G(m)$, approaches the optimal generalization error within $\bigcup F_d$ in the limit of large $m$, and the second term directly penalizes large complexity. If we want the *sum* of the two terms in the bound to be meaningful, then we should be able to give a bound on $\beta(\tilde{d}, m, \delta)$ that decays to 0 with $m$, preferably as rapidly as possible. In other words, *we must be able to argue that the complexity of the hypothesis chosen is limited*. If we can do so, then combined with the bound on the first term we have a proof of the method's *statistical consistency* (that is, approach to the optimal error in the large sample limit), and may even have a nice rate of approach to the optimal error. If we cannot do so, then we are forced to consider the possibility that our method is simply fitting the sample, and incurring large error because as a result. Such a possibility was clearly realized in the experimental results for MDL, where a long period of unbounded hypothesis complexity directly caused a long period of essentially constant generalization error as a function of $m$. We return to this issue after the proof of Theorem 1.

In order to prove Theorem 1, we shall need to following uniform convergence bound which is due to Vapnik [17].

**Uniform Convergence Bound** *Let $F_d$ be a hypothesis class with VC dimension $d < m$. Then, for any given $\delta > 0$, with probability at least $1 - \delta$,*

$$|\epsilon(h) - \hat{\epsilon}(h)| < 2\sqrt{\frac{d\left(\ln\frac{2m}{d} + 1\right) + \ln\frac{9}{\delta}}{m}} \tag{19}$$

*for every $h \in F_d$. If the sample is noisy, then the same bound holds for $\epsilon^\eta(h)$*[10].

**Proof of Theorem 1:** Since $\tilde{d}$ is chosen to minimize $G(\hat{\epsilon}(d), d/m)$, we have that for every $d$

$$G\left(\hat{\epsilon}(\tilde{d}), \tilde{d}/m\right) \leq G\left(\hat{\epsilon}(d), d/m\right). \tag{20}$$

Using the uniform convergence bound stated above we have that for any given $d < m$, with probability at least $1 - \delta/m$,

$$|\epsilon(h) - \hat{\epsilon}(h)| \ < \ 2\sqrt{\frac{d\ln\frac{2em}{d} + \ln\frac{9m}{\delta}}{m}} \tag{21}$$

for all $h \in F_d$. Thus, with probability at least $1-\delta$, the above holds for all $d < m$. For $d \geq m$ we can use the trivial bound that for every $h$, $|\epsilon(h) - \hat{\epsilon}(h)| \leq 1$, and together we have that with probability

[10] In fact, Vapnik (in [17], page 160) gives a more general statement concerning the uniform estimation of probabilities from their frequencies in a class of events of limited VC dimension.

at least $1-\delta$, for every $d$, and for all $h \in F_d$, $|\epsilon(h) - \hat{\epsilon}(h)| < \beta(d, m, \delta)$, where $\beta(\cdot,\cdot,\cdot)$ was defined in the statement of Theorem 1. If we now use the fact that $G(\cdot,\cdot)$ is increasing in its first argument, we can replace the occurrence of $\hat{\epsilon}(\tilde{d})$ on the left-hand side of Equation (20) by $\epsilon(\tilde{d}) - \beta(\tilde{d}, m, \delta)$ to obtain a smaller quantity. Similary, since $\hat{\epsilon}(d) \leq \hat{\epsilon}(h_d)$ (recall that $h_d \stackrel{\text{def}}{=} \text{argmin}_{h \in F_d}\{\epsilon(h)\}$), and $\hat{\epsilon}(h_d) \leq \epsilon(h_d) + \beta(d, m, \delta) = \epsilon_{opt}(d) + \beta(d, m, \delta)$, we can replace the occurrence of $\hat{\epsilon}(d)$ on the right-hand side by $\epsilon_{opt}(d) + \beta(d, m, \delta)$ to obtain a larger quantity. This gives

$$G\left(\epsilon(\tilde{d}) - \beta(\tilde{d}, m, \delta), \tilde{d}/m\right) \leq G\left(\epsilon_{opt}(d) + \beta(d, m, \delta), d/m\right). \tag{22}$$

Now because $G(\cdot,\cdot)$ is an increasing function of its second argument, we can further weaken Equation (22) to obtain

$$G\left(\epsilon(\tilde{d}) - \beta(\tilde{d}, m, \delta), 0\right) \leq G\left(\epsilon_{opt}(d) + \beta(d, m, \delta), d/m\right). \tag{23}$$

If we define $G_0(x) = G(x, 0)$, then since $G(\cdot,\cdot)$ is increasing in its first argument, $G_0^{-1}(\cdot)$ is well-defined, and we may write

$$\epsilon(\tilde{d}) \leq G_0^{-1}\left(G\left(\epsilon_{opt}(d) + \beta(d, m, \delta), d/m\right)\right) + \beta(\tilde{d}, m, \delta). \tag{24}$$

Now fix any small value $\tau > 0$. For this $\tau$, let $d'$ be the smallest value satisfying $\epsilon_{opt}(d') \leq \min_d\{\epsilon_{opt}(d)\} + \tau$ — thus, $d'$ is sufficient complexity to almost match the approximative power of arbitrarily large complexity. Examining the behavior of $G_0^{-1}(G(\epsilon_{opt}(d') + \beta(d', m, \delta), d'/m))$ as $m \to \infty$, we see that the arguments approach the point $(\epsilon_{opt}(d'), 0)$, and so

$$G_0^{-1}(G(\epsilon_{opt}(d') + \beta(d', m, \delta), d'/m)) \longrightarrow G_0^{-1}(G(\epsilon_{opt}(d'), 0)) \tag{25}$$

$$= \epsilon_{opt}(d') \leq \min\{\epsilon_{opt}(d)\} + \tau \tag{26}$$

by continuity of $G(\cdot,\cdot)$, as desired. By defining

$$R_G(m) \stackrel{\text{def}}{=} \min_d \left\{G_0^{-1}\left(G\left(\epsilon_{opt}(d) + \beta(d, m, \delta), d/m\right)\right)\right\} \tag{27}$$

we obtain the statement of the theorem. □

Given the definition of $R_G(m)$ in Equation (27), we can now examine the two terms $R_G(m)$ and $\beta(\tilde{d}, m, \delta)$ more carefully and observe that they may be thought of as competing. In order for $R_G(m)$ to approach $\min_d\{\epsilon_{opt}(d)\}$ *rapidly* and not just asymptotically (that is, in order to have a fast *rate* of convergence), $G(\cdot,\cdot)$ should not penalize complexity too strongly, which is obviously at odds with the optimization of the term $\beta(\tilde{d}, m, \delta)$. For example, consider $G(\hat{\epsilon}(d), d/m) = \hat{\epsilon}(d) + (d/m)^\alpha$ for some power $\alpha > 0$. Assuming $d \leq m$, this rule is conservative (large penalty for complexity) for small $\alpha$, and liberal (small penalty for complexity) for large $\alpha$. Thus, to make $\beta(\tilde{d}, m, \delta)$ small we would like $\alpha$ to be small, to prevent the choice of large $\tilde{d}$. However, by definition of $R_G(m)$ we have that for the function $G$ in question $R_G(m) = \min_d\{\epsilon_{opt}(d) + \beta(d, m, \delta) + (d/m)^\alpha\}$, which increases as $\alpha$ decreases, thus encouraging large $\alpha$ (liberal coding).

Ideally, we might want $G(\cdot,\cdot)$ to balance the two terms of the bound, which implicitly involves finding an appropriately *controlled* but sufficiently *rapid* rate of increase in $\tilde{d}$. The tension between these two criteria in the bound echoes the same tension that was seen experimentally: for MDL, there was a long period of essentially uncontrolled growth of $\tilde{d}$ (linear in $m$), and this uncontrolled growth prevented any significant decay of generalization error (Figures 3 and 4). GRM had controlled growth of $\tilde{d}$, and thus would incur negligible error from our second term — but perhaps this

growth was *too* controlled, as it results in the initially slow (small $m$) decrease in generalization error.

To examine these issues further, we now apply the bound of Theorem 1 to several penalty-based algorithms. In some cases the final form of the bound given in the theorem statement, while easy to interpret, is unnecessarily coarse, and better rates of convergence can be obtained by directly appealing to the proof of the theorem.

We begin with a *simplified* GRM variant (SGRM), defined by $G(\hat{\epsilon}(d), d, m) = \hat{\epsilon}(d) + \beta(d, m, \delta)$. Note that SGRM does not have the exact form required in Theorem 1. However, as we shall show below, its generalization error can be bounded easily using the same techniques applied in the proof of Theorem 1. We first observe that we can avoid weakening Equation (22) to Equation (23), because here $G\left(\epsilon(\tilde{d}) - \beta(\tilde{d}, m, \delta), \tilde{d}, m\right) = \epsilon(\tilde{d})$. Thus the dependence on $\tilde{d}$ in the bound disappears entirely, resulting in the following bound in $\epsilon_{\mathrm{SGRM}}(m)$: With probability at least $1 - \delta$,

$$\epsilon_{\mathrm{SGRM}}(m) \;\leq\; \min_d \{\epsilon_{opt}(d) + 2\beta(d, m, \delta)\}\,. \tag{28}$$

This is not so mysterious, since SGRM penalizes strongly for complexity (even more so than GRM). This bound expresses the generalization error as the minimum of the sum of the best possible error within each class $F_d$ and a penalty for complexity. Such a bound seems entirely reasonable, given that it is essentially the expected value of the empirical quantity we minimized to choose $\tilde{d}$ in the first place. Furthermore, if $\epsilon_{opt}(d) + \beta(d, m, \delta)$ approximates $\epsilon(d)$ well, then such a bound is about the best we could hope for. However, there is no reason in general to expect this to be the case.

As an example of the application of Theorem 1 to MDL we can derive the following bound on $\epsilon_{\mathrm{MDL}}(m)$ (where for any $x > 1$ we define $\mathcal{H}(x)$ to be 1): With probability at least $1 - \delta$,

$$\begin{aligned}
\epsilon_{\mathrm{MDL}}(m) &\leq \min_d \left\{\mathcal{H}^{-1}\left(\mathcal{H}\left(\epsilon_{opt}(d) + \beta(d, m, \delta)\right) + \mathcal{H}(d/m)\right)\right\} + \beta(\tilde{d}_{\mathrm{MDL}}, m, \delta) && (29)\\
&\leq \min_d \{\mathcal{H}\left(\epsilon_{opt}(d) + \beta(d, m, \delta)\right) + \mathcal{H}(d/m)\} + \beta(\tilde{d}_{\mathrm{MDL}}, m, \delta) && (30)\\
&\leq \min_d \{\mathcal{H}(\epsilon_{opt}(d)) + \mathcal{H}(\beta(d, m, \delta)) + \mathcal{H}(d/m)\} + \beta(\tilde{d}_{\mathrm{MDL}}, m, \delta) && (31)\\
&\leq \min_d \{\mathcal{H}(\epsilon_{opt}(d)) + 2\mathcal{H}(\beta(d, m, \delta))\} + \beta(\tilde{d}_{\mathrm{MDL}}, m, \delta) && (32)
\end{aligned}$$

where we have used $\mathcal{H}^{-1}(y) \leq y$ to get Equation (30) and $\mathcal{H}(x + y) \leq \mathcal{H}(x) + \mathcal{H}(y)$ to get Equation (31). Again, we emphasize that the bound given by Equation (32) is vacuous without a bound on $\tilde{d}_{\mathrm{MDL}}$, which we know from the experiments can be of order $m$. However, by combining this bound with an analysis of the behavior of $\tilde{d}_{\mathrm{MDL}}$ for the intervals problem as discussed in Section 4 (see Equation (14) and the discussion following it), it is possible to give an accurate theoretical explanation for the experimental findings for MDL.

As a final example, we apply Theorem 1 to a *variant* of MDL in which the penalty for coding is increased over the original, namely $G(\hat{\epsilon}(d), d/m) = \mathcal{H}(\hat{\epsilon}(d)) + 1/\lambda^2 \mathcal{H}(d/m)$ where $\lambda$ is a parameter that may depend on $d$ and $m$. Assuming that we never choose $\tilde{d}$ whose total penalty is larger than 1 (which holds if we simply add the "fair coin hypothesis" to $F_1$), we have that $\mathcal{H}(\tilde{d}/m) \leq \lambda^2$. Since $\mathcal{H}(x) \geq x$, for all $x$, it follows that $\sqrt{\tilde{d}/m} \leq \lambda$. For any $\delta \geq \exp(-\lambda^2 m)$ we then have that

$$\beta(\tilde{d}, m, \delta) \;<\; 2\sqrt{\frac{\tilde{d}\,\ln(2em) + m\lambda^2 \ln m}{m}} \;=\; O(\lambda\sqrt{\ln m})\,. \tag{33}$$

If $\lambda$ is some decreasing function of $m$ (say, $m^{-\alpha}$ for some $0 < \alpha < 1$), then the bound on $\epsilon(\tilde{d})$ given by Theorem 1 decreases at a reasonable rate.

We conclude this section with two generalizations of Theorem 1. The first is for the case in which the penalty-based algorithm uses a learning algorithm $L$ which does not necessarily minimize the training error, and the second is for the case in which the sample is corrupted by noise.

For Theorem 2 we need the following definition. We say that a learning algorithm $L$ is *adequate* if it has the following property. There exists a function $\mu_L : \mathcal{N} \times \mathcal{N} \times [0,1] \to [0,1]$, such that for every given $\delta$, with probability at least $1-\delta$, $|\hat{\epsilon}_L(d) - \hat{\epsilon}_{opt}(d)| \leq \mu_L(d,m,\delta)$ for all $d$, where $\hat{\epsilon}_{opt} \stackrel{\text{def}}{=} \min_{h \in F_d}\{\hat{\epsilon}(h)\}$. That is, $\hat{\epsilon}_{opt}$ is the minimum training error (on the sample $S$) achievable in $F_d$. Furthermore, as $m \to \infty$, $\mu_L(d,m,\delta) \to \bar{\mu}_L$, where $\bar{\mu}_L$ is some constant which depends on $L$. Thus, if $\bar{\mu}_L$ is not very large, then in the limit of large $m$, $L$ does not perform much worse than the training error minimization algorithm. We would like to note that many other definitions of adequacy are appropriate, and can lead to statements similar to the one in Theorem 2 below.

**Theorem 2** *Let $(\{F_d\}, f, D, L)$ be an instance of the model selection problem in which $L$ is an adequate learning algorithm, and where $d$ is the VC dimension of $F_d$. Let $G : [0,1] \times \Re \to \Re$ be a function that is continuous and increasing in both its arguments, and let $\epsilon_G(m)$ denote the generalization error of the penalty-based model selection algorithm $\tilde{d} = \text{argmin}_d\{G(\hat{\epsilon}(d), d/m)\}$ on a training sample of size $m$. Then*

$$\epsilon_G(m) \leq R_G(m) + \beta(\tilde{d}, m, \delta/2) \tag{34}$$

*where $R_G(m)$ approaches $\min_d\{\epsilon_{opt}(d)\} + \bar{\mu}_L$ as $m \to \infty$, and is defined as follows:*

$$R_G(m) \stackrel{\text{def}}{=} \min_d \left\{ G_0^{-1}\left(G\left(\epsilon_{opt}(d) + \mu_L(d,m,\delta/2) + \beta(d,m,\delta/2)\right), d/m\right)\right\} \tag{35}$$

*where $G_0(\cdot) \stackrel{\text{def}}{=} G(\cdot, 0)$.*

**Proof Sketch:** The proof is very similar to the proof of Theorem 1 and hence we need only point out the differences. As in the proof of Theorem 1 we have that for any value of $d$ $G\left(\hat{\epsilon}(\tilde{d}), \tilde{d}/m\right) \leq G\left(\hat{\epsilon}(d), d/m\right)$. It is still true that with probability at least $1-\delta/2$, $\hat{\epsilon}(\tilde{d})$ is bounded from below by $\epsilon(\tilde{d}) + \beta(\tilde{d}, m, \delta/2)$, however, we cannot bound $\hat{\epsilon}(d)$ by $\epsilon_{opt}(d) + \beta(d,m,\delta/2)$ since it is not true any longer that $\hat{\epsilon}(d)$ is the minimal error achievable in $F_d$. Instead we have that with probability at least $1-\delta/2$, for every $d$, $\hat{\epsilon}(d) \leq \hat{\epsilon}_{opt}(d) + \mu(d,m,\delta/2)$, and hence with probability at least $1-\delta$, $\hat{\epsilon}(d) \leq \epsilon_{opt}(d) + \mu(d,m,\delta/2) + \beta(d,m,\delta/2)$. The rest of the proof follows as in Theorem 1 where we get that for every $d$

$$\epsilon(\tilde{d}) \;\leq\; G_0^{-1}\left(G\left(\epsilon_{opt}(d) + \mu(d,m,\delta/2) + \beta(d,m,\delta/2), d/m\right)\right) + \beta(\tilde{d}, m, \delta/2). \tag{36}$$

Using our assumption on the adequacy of $L$ we have that as $m \to \infty$,

$$\min_d \left\{ G_0^{-1}\left(G\left(\epsilon_{opt}(d) + \mu(d,m,\delta/2) + \beta(d,m,\delta/2), d/m\right)\right)\right\} \to \min\{\epsilon_{opt}(d)\} + \bar{\mu}_L, \tag{37}$$

as required. □

**Theorem 3** *Let $(\{F_d\}, f, D, L)$ be an instance of the model selection problem in which $L$ performs training error minimization, and where $d$ is the VC dimension of $F_d$. Let $G : [0,1] \times \Re \to \Re$ be a function that is continuous and increasing in both its arguments, and let $\epsilon_G(m)$ denote the expected*

*generalization error of the penalty-based model selection algorithm* $\tilde{d} = \mathrm{argmin}_d\{G(\hat{\epsilon}(d), d/m)\}$ *on a training sample of size* $m$*, and in the presence of noise at rate* $\eta$*. Then*

$$\epsilon_G(m) \leq R_G(m,\eta) + \frac{1}{1-2\eta}\beta(\tilde{d}, m, \delta) \tag{38}$$

*where* $R_G(m,\eta)$ *approaches* $\min_d\{\epsilon_{opt}(d)\}$ *as* $m \to \infty$*, and is defined as follows:*

$$R_G(m,\eta) \stackrel{\mathrm{def}}{=} \frac{1}{1-2\eta}\min_d\left\{G_0^{-1}\left(G\left((1-2\eta)\epsilon_{opt}(d) + \eta + \beta(d,m,\delta)\right), d/m\right)\right\} - \eta \tag{39}$$

*where* $G_0(\cdot) \stackrel{\mathrm{def}}{=} G(\cdot, 0)$.

**Proof Sketch:** The proof of Theorem 3 follows the same sequence of inequalities as the proof of Theorem 1, except that each occurence of $\epsilon(\cdot)$ should be exchanged with $\epsilon^\eta(\cdot)$, and each occurence of $\epsilon_{opt}(\cdot)$ should be exchanged with $\epsilon^\eta_{opt}(\cdot)$. Thus, similary to Equation (24), we have that

$$\epsilon^\eta(\tilde{d}) \;\leq\; G_0^{-1}\left(G\left(\epsilon^\eta_{opt}(d) + \beta(d,m,\delta), d/m\right)\right) + \beta(\tilde{d}, m, \delta). \tag{40}$$

If we now apply the equality $\epsilon^\eta(h) = (1-2\eta)\epsilon(h) + \eta$ we get that for every $d$

$$\epsilon(\tilde{d}) \;\leq\; \frac{1}{1-2\eta}\left[G_0^{-1}\left(G\left((1-2\eta)\epsilon_{opt}(d) + \eta + \beta(d,m,\delta), d/m\right)\right) + \beta(\tilde{d}, m, \delta)\right] - \eta. \tag{41}$$

Again, similarly to the proof of Theorem 1, we have that as $m \to \infty$,

$$\min_d\left\{G_0^{-1}\left(G\left((1-2\eta)\epsilon_{opt}(d) + \eta + \beta(d,m,\delta), d/m\right)\right)\right\} \to (1-2\eta)\min\{\epsilon_{opt}(d)\} + \eta, \tag{42}$$

and thus we get the desired bound. Note that the rate of convergence of of $R_G(m,\eta)$ to the optimal error depends now on the size of $\eta$ as well as on $G$. The same is true for the penalty complexity term in the bound. It is not very surprising, that as $\eta$ approaches $1/2$, the bound worsens. □

# 6 A Bound on the Additional Error of CV

In this section we state a general theorem bounding the additional generalization error suffered by cross validation compared to any *polynomial complexity* model selection algorithm $M$. By this we mean that given a sample of size $m$, algorithm $M$ will never choose a value of $\tilde{d}$ larger than $m^k$ for some fixed exponent $k > 1$. We emphasize that this is a mild condition that is met in practically every realistic model selection problem: although there are many documented circumstances in which we may wish to choose a model whose complexity is on the order of the sample size, we do not imagine wanting to choose, for instance, a neural network with a number of nodes *exponential* in the sample size.

**Theorem 4** *Let* $M$ *be any polynomial complexity model selection algorithm, and let* $(\{F_d\}, f, D, L)$ *be any instance of model selection. Let* $\epsilon_{\mathrm{M}}(m)$ *and* $\epsilon_{\mathrm{CV}}(m)$ *denote the expected generalization error of the hypotheses chosen by* $M$ *and CV respectively. Then for any given* $\delta > 0$*, with probability at least* $1-\delta$*:*

$$\epsilon_{\mathrm{CV}}(m) \leq \epsilon_{\mathrm{M}}((1-\gamma)m) + O\left(\sqrt{\frac{\ln(m/\delta)}{\gamma m}}\right). \tag{43}$$

*In other words, the generalization error of CV on* $m$ *examples is at most the generalization error of* $M$ *on* $(1-\gamma)m$ *examples, plus the "test penalty term"* $O(\sqrt{\ln(m/\delta)/(\gamma m)})$.

**Proof:** Let $S = (S', S'')$ be a random sample of $m$ examples, where $|S'| = (1-\gamma)m$ and $|S''| = \gamma m$. Let $d_{max} = ((1-\gamma)m)^k$ be the polynomial bound on the complexity selected by $M$, and let $\tilde{h}'_1 \in F_1, \ldots, \tilde{h}'_{d_{max}} \in F_{d_{max}}$ be determined by $\tilde{h}'_d = L(S', d)$. By definition of CV, $\tilde{d}$ is chosen according to $\tilde{d} = \mathrm{argmin}_d\{\hat{\epsilon}_{S''}(\tilde{h}'_d)\}$. For a given $\tilde{h}'_d$, we know by Hoeffding's Inequality [5] that for any $\alpha > 0$,

$$Pr\left[\left|\epsilon(\tilde{h}'_d) - \hat{\epsilon}_{S''}(\tilde{h}'_d)\right| > \alpha\right] < 2\exp(-2\alpha^2\gamma m). \tag{44}$$

The probability that some $\tilde{h}'_d$ deviates by more than $\alpha$ from its expected value is therefore bounded by $2m^k \exp(-2\alpha^2\gamma m)$. It follows that for any given $\delta$, with probability at least $1-\delta$ over the draw of $S''$,

$$|\epsilon(\tilde{h}'_d) - \hat{\epsilon}_{S''}(\tilde{h}'_d)| = O\left(\sqrt{\frac{\ln(m/\delta)}{\gamma m}}\right) \tag{45}$$

for all $d \leq d_{max}$. Therefore with probability at least $1-\delta$

$$\epsilon_{\mathrm{CV}} = \min_d\{\epsilon(\tilde{h}'_d)\} + O\left(\sqrt{\frac{\ln(m/\delta)}{\gamma m}}\right). \tag{46}$$

But as we have previously observed, the generalization error of *any* model selection algorithm (including $M$) on input $S'$ is lower bounded by $\min_d\{\epsilon(\tilde{h}'_d)\}$, and our claim directly follows. □

Note that the bound of Theorem 4 does *not* claim $\epsilon_{\mathrm{CV}}(m) \leq \epsilon_{\mathrm{M}}(m)$ for all M (which would mean that cross validation is an optimal model selection algorithm). The bound given is weaker than this ideal in two important ways. First, and perhaps most importantly, $\epsilon_{\mathrm{M}}((1-\gamma)m)$ may be considerably larger than $\epsilon_{\mathrm{M}}(m)$. This could either be due to properties of the underlying learning algorithm $L$, or due to inherent *phase transitions* (sudden decreases) in the optimal information-theoretic learning curve [14, 4] — thus, in an extreme case, it could be that the generalization error that can be achieved within some class $F_d$ by training on $m$ examples is close to 0, but that the optimal generalization error that can be achieved in $F_d$ by training on a slightly smaller sample is near 1/2. This is intuitively the worst case for cross validation — when the small fraction of the sample saved for testing was critically needed for training in order to achieve nontrivial performance — and is reflected in the first term of our bound. Obviously the risk of "missing" phase transitions can be minimized by decreasing the test fraction $\gamma$, but only at the expense of increasing the test penalty term, which is the second way in which our bound falls short of the ideal. However, unlike the potentially unbounded difference $\epsilon_{\mathrm{M}}((1-\gamma)m) - \epsilon_{\mathrm{M}}(m)$, our bound on the test penalty can be decreased without any problem-specific knowledge by simply *increasing* the test fraction $\gamma$.

Despite these two competing sources of additional CV error, the bound has some strengths that are worth discussing. First of all, the bound does not simply compare the worst-case error of CV to the worst-case error of $M$ over a wide class of model selection problems; the bound holds for *any* model selection problem instance $(\{F_d\}, f, D, L)$. We believe that giving similarly general bounds for any penalty-based algorithm would be extremely difficult, if not impossible. The reason for this belief arises from the diversity of learning curve behavior documented by the statistical mechanics approach [14, 4], among other sources. In the same way that there is no universal learning curve behavior, there is no universal behavior for the relationship between the functions $\hat{\epsilon}(d)$ and $\epsilon(d)$ — the relationship between these quantities may depend critically on the target function and the input distribution (this point is made more formally in Section 7). CV is sensitive to this dependence by virtue of its target function-dependent and distribution-dependent estimate of $\epsilon(d)$. In contrast, by their very nature, penalty-based algorithms propose a *universal* penalty to be assigned to the observation of error $\hat{\epsilon}(h)$ for a hypothesis $h$ of complexity $d$.

A more technical feature of Theorem 4 is that it can be combined with bounds derived for penalty-based algorithms using Theorem 1 to suggest how the parameter $\gamma$ should be tuned. For example, letting $M$ be the SGRM algorithm described in Section 5, and combining Equation (28) with Theorem 4 yields

$$\epsilon_{\mathrm{CV}}(m) \leq \epsilon_{\mathrm{SGRM}}((1-\gamma)m) + 2\sqrt{\ln(d_{\mathrm{MAX}}(m)/\delta)/\gamma m} \quad (47)$$

$$\leq \min_d \{\epsilon_{opt}(d) + 2\beta(d,(1-\gamma)m,\delta)\} + 2\sqrt{\frac{\ln(d_{\mathrm{MAX}}(m)/\delta)}{\gamma m}} \quad (48)$$

If we knew the form of $\epsilon_{opt}(d)$ (or even had bounds on it), then in principle we could minimize the bound of Equation (48) as a function of $\gamma$ to derive a recommended training/test split. Such a program is feasible for many specific problems (such as the intervals problem), or by investigating general but plausible bounds on the approximation rate $\epsilon_{opt}(d)$, such as $\epsilon_{opt}(d) \leq c_0/d$ for some constant $c_0 > 0$. For a detailed study of this line of inquiry, see Kearns [6]. Here we simply note that Equation (48) tells us that in cases for which the power law decay of generalization error within each $F_d$ holds approximately, the performance of CV will be competitive with GRM or any other algorithm. This makes perfect sense in light of the preceding analysis of the two sources for additional CV error: in problems with power law learning curve behavior, we have a power law bound on $\epsilon_{\mathrm{M}}((1-\gamma)m) - \epsilon_{\mathrm{M}}(m)$, and thus CV "tracks" any other algorithm closely in terms of generalization error. This is exactly the behavior observed in the experiments described in Section 4, for which the power law is known to hold approximately.

We conclude this section with a noisy version of Theorem 4, whose correctness directly follows from the proof of Theorem 4, together with the equality $\epsilon^{\eta}(h) = (1-2\eta)\epsilon(h) + \eta$.

**Theorem 5** *Let $M$ be any polynomial complexity model selection algorithm, and let $(\{F_d\}, f, D, L)$ be any instance of model selection. Let $\epsilon_{\mathrm{M}}(m)$ and $\epsilon_{\mathrm{CV}}(m)$ denote the expected generalization error of the hypotheses chosen by $M$ and CV respectively when the sample is corrupted by noise at rate $\eta$. Then for any given $\delta > 0$, with probability at least $1-\delta$*

$$\epsilon_{\mathrm{CV}}(m) \leq \epsilon_{\mathrm{M}}((1-\gamma)m) + O\left(\frac{1}{1-2\eta}\sqrt{\ln(m/\delta)/(\gamma m)}\right). \quad (49)$$

# 7 Limitations on Penalty-Based Algorithms

Recall that our experimental findings suggested that it may sometimes be fair to think of penalty-based algorithms as being either conservative or liberal in the amount of coding they are willing to allow in their hypothesis, and that bias in either direction can result in suboptimal generalization that is not easily overcome by tinkering with the form of the rule. In this section we treat this intuition more formally, by giving a theorem demonstrating some fundamental limitations on the diversity of problems that can be effectively handled by any fixed penalty-based algorithm. Briefly, we show that there are (at least) two very different forms that the relationship between $\hat{\epsilon}(d)$ and $\epsilon(d)$ can assume, and that any penalty-based algorithm can perform well on only one of these. Furthermore, for the problems we choose, CV can in fact succeed on both. Thus we are doing more than simply demonstrating that no model selection algorithm can succeed universally for all target functions, a statement that is intuitively obvious and has been made more formal in recent papers [13, 19]. We are in fact identifying a weakness that is *special* to penalty-based algorithms. However, as we have discussed previously, the use of CV is not without pitfalls of its own. We

therefore conclude the paper in Section 8 with a summary of the different risks involved with each type of algorithm, and a discussion of our belief that in the absence of detailed problem-specific knowledge, our overall analysis favors the use of CV.

**Theorem 6** *For any sample size $m$, there are model selection problem instances $(\{F_1^d\}, f_1, D_1, L)$ and $(\{F_2^d\}, f_2, D_2, L)$ (where $L$ performs empirical error minimization in both instances) and a constant $\lambda$ independent of $m$ such that for any penalty-based model selection algorithm $G$, either $\epsilon_1^G(m) \geq \min_d\{\epsilon_1(d)\} + \lambda$ or $\epsilon_2^G(m) \geq \min_d\{\epsilon_2(d)\} + \lambda$. Here $\epsilon_i(d)$ is the function $\epsilon(d)$ for instance $i \in \{1,2\}$, and $\epsilon_i^G(m)$ is the expected generalization error of algorithm $G$ for instance $i$. Thus, on at least one of the two model selection problems, the generalization error of $G$ is lower bounded away from the optimal value $\min_d\{\epsilon_i(d)\}$ by a constant independent of $m$.*

**Proof**: For notational convenience, in the proof we use $\hat{\epsilon}_i(d)$ and $\epsilon_i(d)$ ($i \in \{1,2\}$) to refer to the expected values of these functions. We start with a rough description of the properties of the two problems (see Figure 26): in Problem 1, the "right" choice of $d$ is 0, any additional coding directly results in larger generalization error, and the training error, $\hat{\epsilon}_1(d)$, decays gradually with $d$. In Problem 2, a large amount of coding is required to achieve nontrivial generalization error, and the training error remains large as $d$ increases until $d = m/2$, where the training error drops rapidly.

More precisely, we will arrange things so that the first model selection problem (Problem 1) has the following properties

1. The function $\hat{\epsilon}_1(d)$ lies above the linear function $f(d) = \eta_1(1-\eta_1) - d/(2m)$ whose $y$-intercept is $\eta_1(1-\eta_1)$, and whose $x$-intercept is $2\eta_1(1-\eta_1)m \leq m/2$;
2. $\epsilon_1(d)$ is minimized at $d = 0$, and furthermore, for any constant $c$ we have $\epsilon_1(cm) \geq c/2$.

We will next arrange that the second model selection problem (Problem 2) will obey:

1. The function $\hat{\epsilon}_2(d) = a_1$ for $0 \leq d \leq 2\eta_1(1-\eta_1)m \leq m/2$, where $\eta_1(1-\eta_1) > a_1$;
2. The function $\epsilon_2(d)$ is lower bounded by $a_1$ for $0 \leq d < m/2$, but $\epsilon_2(m/2) = 0$.

In Figure 26 we illustrate the conditions on $\hat{\epsilon}(d)$ for the two problems, and also include hypothetical instances of $\hat{\epsilon}_1(d)$ and $\hat{\epsilon}_2(d)$ that are consistent with these conditions (and are furthermore representative of the "true" behavior of the $\hat{\epsilon}(d)$ functions actually obtained for the two problems we define in a moment).

We can now give the underlying logic of the proof using the hypothetical $\hat{\epsilon}_1(d)$ and $\hat{\epsilon}_2(d)$. Let $\tilde{d}_1$ denote the complexity chosen by $G$ for Problem 1, and let $\tilde{d}_2$ be defined similarly. First consider the behavior of $G$ on Problem 2. In this problem we know by our assumptions on $\epsilon_2(d)$ that if $G$ fails to choose $\tilde{d}_2 \geq m/2$, $\epsilon_2^G \geq a_1$, already giving a constant lower bound on $\epsilon_2^G$ for this problem. This is the easier case; thus let us assume that $\tilde{d}_2 \geq m/2$, and consider the behavior of $G$ on Problem 1. Referring to Figure 26, we see that for $0 \leq d \leq d_0$ (where $d_0$ is such that $\hat{\epsilon}_1(d_0) = a_1$), $\hat{\epsilon}_1(d) \geq \hat{\epsilon}_2(d)$, and thus

$$\text{For } 0 \leq d \leq d_0, \;\; G(\hat{\epsilon}_1(d), d/m) \geq G(\hat{\epsilon}_2(d), d/m) \tag{50}$$

(because penalty-based algorithms assign greater penalties for greater training error or greater complexity). Since we have assumed that $\tilde{d}_2 \geq m/2$, we know that

$$\text{For } d < m/2, \;\; G(\hat{\epsilon}_2(d), d/m) \geq G(\hat{\epsilon}_2(\tilde{d}), \tilde{d}_2/m) \tag{51}$$

and in particular, this inequality holds for $0 \leq d \leq d_0$. On the other hand, by our choice of $\hat{\epsilon}_1(d)$, $\hat{\epsilon}_1(\tilde{d}_2) = 0$ (and thus $\hat{\epsilon}_2(\tilde{d}_2) \geq \hat{\epsilon}_1(\tilde{d}_2)$). Therefore,

$$G(\hat{\epsilon}_2(\tilde{d}_2), \tilde{d}_2/m) \geq G(\hat{\epsilon}_1(\tilde{d}_2), \tilde{d}_2/m) \ . \tag{52}$$

Combining the three inequalities above (Equations (50), (51) and (52)), we have that

$$\text{For } 0 \leq d \leq d_0, \ \ G(\hat{\epsilon}_1(d), d/m) \geq G(\hat{\epsilon}_1(\tilde{d}_2), \tilde{d}_2/m) \tag{53}$$

from which it directly follows that in Problem 1, $G$ *cannot* choose $0 \leq \tilde{d}_1 \leq d_0$. From the definition of $f(d)$ in our first condition on Problem 1, it follows that $d_0 \geq 2(\eta(1-\eta) - a_1)\ m$. Using the second condition on Problem 1 we get that $\epsilon_1^G \geq \epsilon_1(d_0) \geq \eta(1-\eta) - a_1$, and thus we have a constant lower bound on $\epsilon_1^G$.

Now we describe the two problems used in the proof, and briefly argue why they have the desired properties; We are in fact already familiar with the first problem: the class $F_1^d$ is simply the class of all $d$-alternation functions over $[0,1]$, the target function is the 0-alternation function that classifies all examples as negative, the input distribution $D_1$ is uniform over $[0,1]$, and we may choose any constant noise rate $\eta_1$. Now clearly under these settings we have $\epsilon_1^{opt}(0) = \epsilon_1(0) = 0$ (where for convenience of notation we let $\epsilon_i^{opt}(d)$ denote $\epsilon_{opt}(d)$ for problem $i$), and $\epsilon_1(d) > 0$ for any $d > 0$ (because the noise in the sample will cause us to code "false alternations"). Furthermore, each additional false interval that is coded will result in an additional $\Theta(1/m)$ generalization error, thus resulting in the desired property $\epsilon_1(cm) \geq c/2$. Finally, we obviously expect $\hat{\epsilon}_1(0) = \eta_1$, and using the same argument applied in the explanation of Equation (14) (where in our case $s = 0$), we have that the expected number of label alternations required to achieve training error 0 is $2\eta_1(1-\eta_1)m$. Furthermore, for every $d < 2\eta_1(1-\eta_1)m$, $\hat{\epsilon}_1(d+2) \leq \hat{\epsilon}_1(d) - 1/m$ (since by adding two switch points, at least one additional sample point can be labeled consistently). Hence, $\hat{\epsilon}_1(d)$ must lie above the linear function whose slope is $-1/(2m)$ and whose $x$-intercept is $2\eta_1(1-\eta_1)m$, as required.

For the second problem, let us begin with the input space $\{0,1\}^N$ for some value $N >> m$. The function class $F_2^d$ consists of all parity functions in which only the variables $x_1, \ldots, x_d$ are permitted to appear, the target function $f \in F_2^{m/2}$ is $f(\vec{x}) = x_1 \oplus \cdots \oplus x_{m/2}$, and the input distribution $D_2$ is uniform over $\{0,1\}^N$. The noise rate $\eta_2 = 0$ (larger values will work as well). Under these settings, it holds (since the distance between every two different parity functions is $1/2$) that $\epsilon_2^{opt}(d) = 1/2$ for $0 \leq d < m/2$, thus implying that $\epsilon_2(d) \geq 1/2$ in the same range. Furthermore, since $f \in F_2^{m/2}$, $\epsilon_2^{opt}(m/2) = 0$ and with high probability (for a large enough sample) $\epsilon_2(m/2) = 0$ and $\hat{\epsilon}_2(d) \approx 1/2$ for $0 \leq d < m/2$. Note that we have almost satisfied the desired conditions on Problem 2, using the value $a_1 = 1/2$; however, the conditions on Problem 2 and the lower bound argument given above require further that $\eta_1(1-\eta_1) > a_1$. We can easily arrange this final condition by simply scaling down $a_1$, by adding a "special" point to the domain on which all functions in $F_2^d$ agree (details are omitted). Referring to Figure 26, notice that the "optimal" setting of $a_1$ is determined by the trade-off between $a_1$ (which lower bounds the error of algorithms failing on Problem 2) and $d_0/m$ (which lower bounds the error of algorithms failing on Problem 1). This concludes the proof for Theorem 6. □

We note that although Theorem 6 was designed to create two model selection problems with the most disparate behavior possible, the proof technique can be used to give lower bounds on the generalization error of penalty-based algorithms under more general settings. Finally, we remark that Theorem 6 can be strengthened to hold for a *single* model selection problem (that is, a single function class sequence and distribution), with only the target function changing to obtain the two different behaviors. This rules out the salvation of the penalty-based algorithms via problem-specific parameters to be tuned, such as "effective dimension".

# 8 Conclusions

Based on both our experimental and theoretical results, we offer the following conclusions:

Model selection algorithms that attempt to reconstruct the curve $\epsilon(d)$ solely by examining the curve $\hat{\epsilon}(d)$ often have a tendency to overcode or undercode in their hypothesis for small sample sizes, which is exactly the sample size regime in which model selection is an issue. Such tendencies are not easily eliminated without suffering the reverse tendency.

There exist model selection problems in which a hypothesis whose complexity is close to the sample size should be chosen, and in which a hypothesis whose complexity is close to 0 should be chosen, but that generate $\hat{\epsilon}(d)$ curves with insufficient information to distinguish which is the case. The penalty-based algorithms cannot succeed in both cases, whereas CV can.

The error of CV can be bounded in terms of the error of any other algorithm. The only cases in which the CV error may be dramatically worse are those in which phase transitions occur in the underlying learning curves at a sample size larger than that held out for training by CV.

Thus we see that both types of algorithms considered have their own Achilles' Heel. For penalty-based algorithms, it is an inability to distinguish two types of problems that call for drastically different hypothesis complexities. For CV, it is phase transitions that unluckily fall between $(1-\gamma)m$ examples and $m$ examples. On balance, we feel that the evidence we have gathered favors use of CV in most common circumstances. Perhaps the best way of stating our position is as follows: given the general upper bound on CV error we have obtained, and the limited applicability of any fixed penalty-based rule demonstrated by Theorem 6 and the experimental results, the burden of proof lies with the practitioner who favors an penalty-based algorithm over CV. In other words, such a practitioner should have concrete evidence (experimental or theoretical) that their algorithm will outperform CV on the problem of interest. Such evidence *must* arise from detailed problem-specific knowledge, since we have demonstrated here the diversity of behavior that is possible in natural model selection problems.

Finally, we wish to remark that although we have limited our attention here to the case of supervised learning of boolean functions, we believe that many of the principles uncovered (such as the limitations of penalty-based algorithms, and the tracking abilities of cross validation) will be applicable to practically any learning setting in which there is a model minimizing an expected loss (generalization error) must be derived from independent observations from a source. A prime example for further investigation would be distribution learning with respect to the Kullback-Liebler divergence (log loss), where $\epsilon_{opt}$-based upper bounds for MDL-like rules are already known [1], yet there also exist phase transitions for natural problems [4].

### Acknowledgements

We give warm thanks to Yoav Freund and Ronitt Rubinfeld for their collaboration on various portions of the work presented here, and for their insightful comments. Thanks to Sebastian Seung and Vladimir Vapnik for interesting and helpful conversations.

## References


[1] A. R. Barron and T. M. Cover. Minimum complexity density estimation. *IEEE Transactions on Information Theory*, 37:1034–1054, 1991.

[2] A. Blum and R. L. Rivest. Training a 3-node neural net is NP-Complete. In David S. Touretzky, editor, *Advances in Neural Information Processing Systems I*, pages 494–501. Morgan Kaufmann, San Mateo, CA, 1989.

[3] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.

[4] D. Haussler, M. Kearns, H.S. Seung, and N. Tishby. Rigourous learning curve bounds from statistical mechanics. In *Proceedings of the Seventh Annual ACM Confernce on Computational Learning Theory*, pages 76–87, 1994.

[5] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, March 1963.

[6] M. Kearns. A bound on the error of cross validation, with consequences for the training-test split. In *Advances in Neural Information Processing Systems 8*. The MIT Press, 1996. To appear.

[7] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. In *Proceedings of the 5th Annual Workshop on Computational Learning Theory*, pages 341–352. ACM Press, New York, NY, 1992.

[8] L. Pitt and L. Valiant. Computational limitations on learning from examples. *Journal of the ACM*, 35:965–984, 1988.

[9] J. R. Quinlan and R. L. Rivest. Inferring decision trees using the minimum description length principle. *Information and Computation*, 80(3):227–248, 1989.

[10] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[11] J. Rissanen. Stochastic complexity and modeling. *Annals of Statistics*, 14(3):1080–1100, 1986.

[12] J. Rissanen. *Stochastic Complexity in Statistical Inquiry*, volume 15 of *Series in Computer Science*. World Scientific, 1989.

[13] C. Schaffer. A conservation law for generalization performance. In *Proceedings of the Eleventh International Conference on Machine Learning*, pages 259–265, 1994.

[14] H. S. Seung, H. Sompolinsky, and N. Tishby. Statistical mechanics of learning from examples. *Physical Review*, A45:6056–6091, 1992.

[15] M. Stone. Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society B*, 36:111–147, 1974.

[16] M. Stone. Asymptotics for and against cross-validation. *Biometrika*, 64(1):29–35, 1977.

[17] V. N. Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer-Verlag, New York, 1982.

[18] V. N. Vapnik and A. Y. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and its Applications*, 16(2):264–280, 1971.

[19] D. H. Wolpert. On the connection between in-sample testing and generalization error. *Complex Systems*, 6:47–94, 1992.

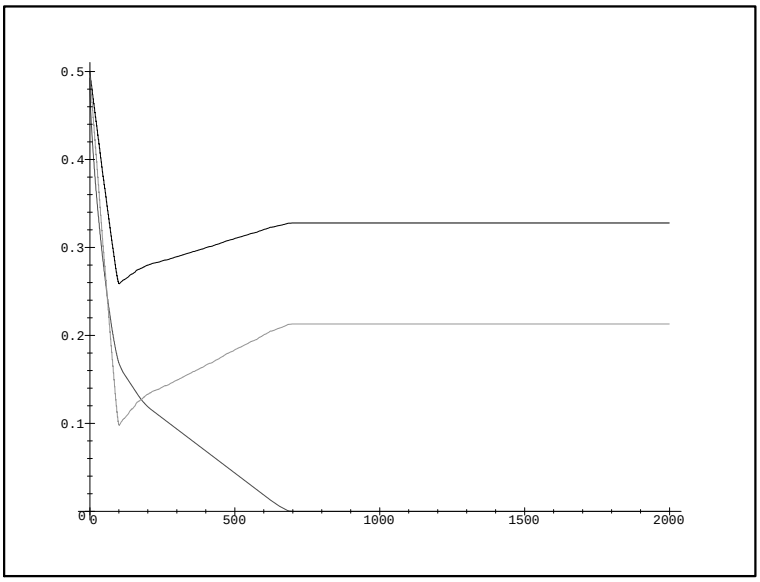


Figure 1: Experimental plots of the functions $\epsilon(d)$ (lower curve with local minimum), $\epsilon^{\eta}(d)$ (upper curve with local minimum) and $\hat{\epsilon}(d)$ (monotonically decreasing curve) versus complexity $d$ for a target function of 100 alternating intervals, sample size 2000 and noise rate $\eta = 0.2$. Each data point represents an average over 10 trials. The flattening of $\epsilon(d)$ and $\epsilon^{\eta}(d)$ occurs at the point where the noisy sample can be realized with no training error. ; by convention, our algorithm never adds more alternations of label than necessary to achieve zero training error. Note that the Vapnik model of $\epsilon(d)$ as the sum of $\hat{\epsilon}(d)$ plus a complexity penalty term of the approximate form $\sqrt{d/m}$ is fairly accurate here; see Figure 2.

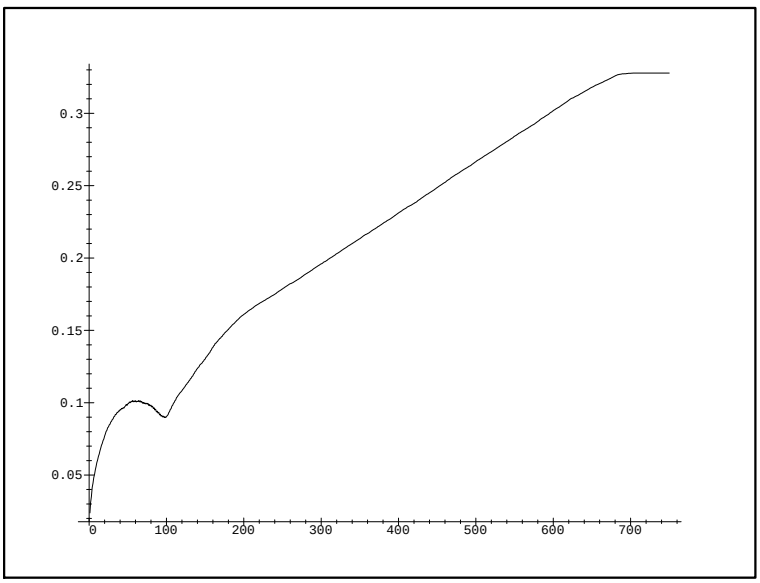


Figure 2: Plot of $\epsilon(d) - \hat{\epsilon}(d)$ versus complexity $d$ for the same experiments used to obtain Figure 1. As function of $d/m$ it appears that $\hat{\epsilon}(d) - \epsilon(d)$ has an initial regime (for $d << 100$, or for this $m$, $d/m < 100/2000 = 0.05$) with behavior that is approximately $\Theta(\sqrt{d/m})$, and a later regime (for $d/m >> 0.05$) in which the behavior is linear in $d/m$.

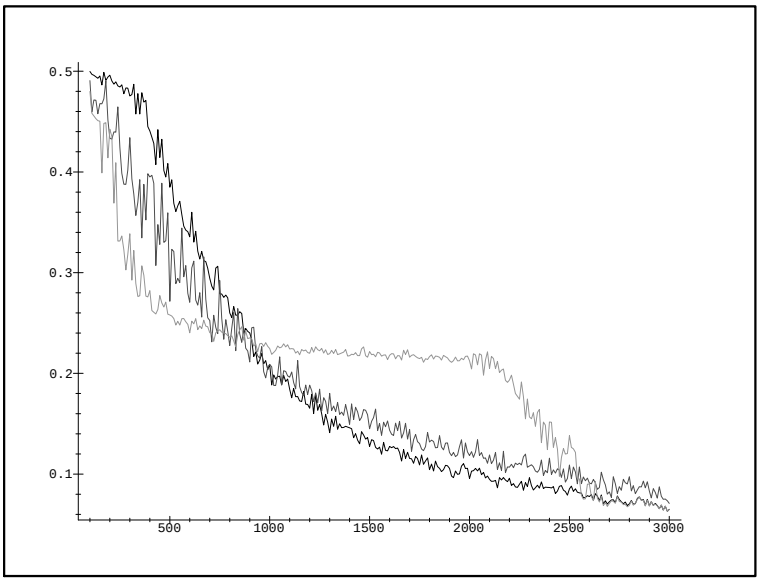


Figure 3: Experimental plots of generalization errors $\epsilon_{\mathrm{MDL}}(m)$ (most rapid initial decrease), $\epsilon_{\mathrm{CV}}(m)$ (intermediate initial decrease) and $\epsilon_{\mathrm{GRM}}(m)$ (least rapid initial decrease) versus sample size $m$ for a target function of 100 alternating intervals and noise rate $\eta = 0.20$. Each data point represents an average over 10 trials. Note that the "shelf" of $\epsilon_{\mathrm{MDL}}$ is approximately at the noise rate $\eta = 0.20$, since MDL is coding all the noisy labels. Also, note that the performance of MDL relative to the other two methods is degrading as the noise rate increases.

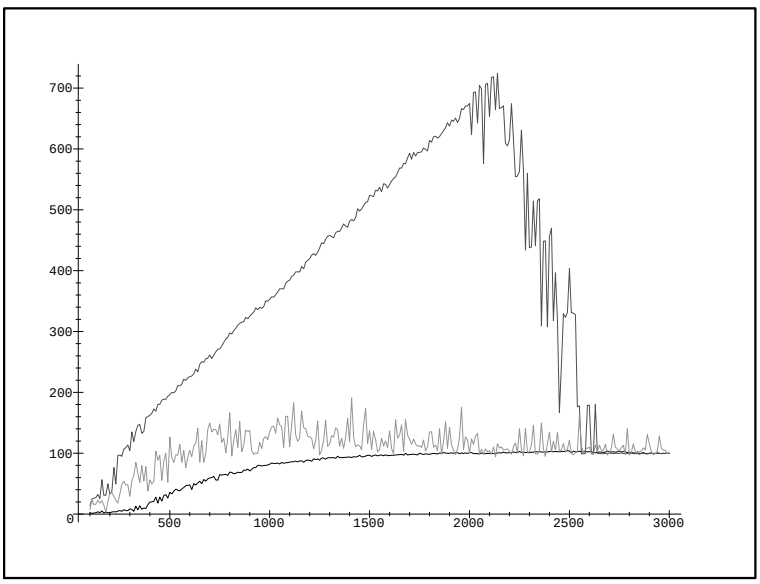


Figure 4: Experimental plots of hypothesis lengths $\tilde{d}_{\mathrm{MDL}}(m)$ (most rapid initial increase), $\tilde{d}_{\mathrm{CV}}(m)$ (intermediate initial increase) and $\tilde{d}_{\mathrm{GRM}}(m)$ (least rapid initial increase) versus sample size $m$ for a target function of 100 alternating intervals and noise rate $\eta = 0.20$. Each data point represents an average over 10 trials.

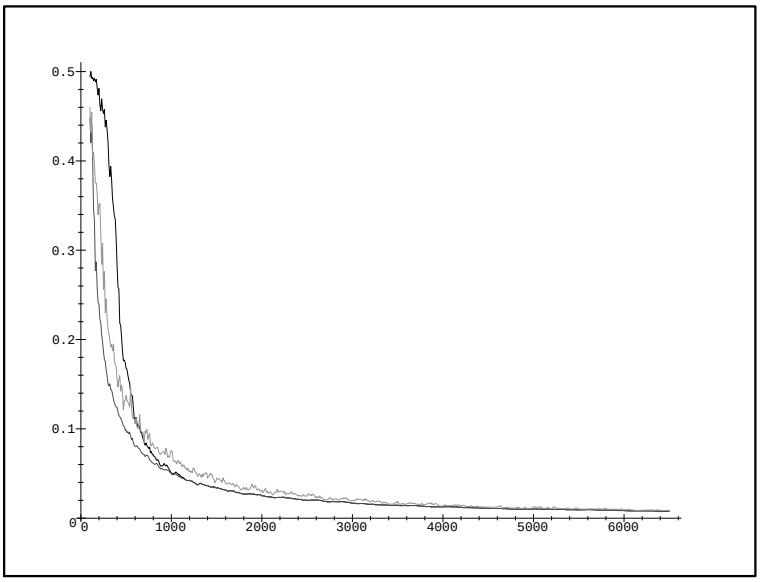


Figure 5: Experimental plots of generalization errors $\epsilon_{\mathrm{MDL}}$ (most rapid initial decrease), $\epsilon_{\mathrm{CV}}$ (intermediate initial decrease) and $\epsilon_{\mathrm{GRM}}$ (least rapid initial decrease) as a function of sample size for a target function of 100 alternating intervals and noise rate $\eta = 0.0$. Each data point represents an average over 10 trials. Note the similar performance for the three methods in this noise-free case, where there is no danger of "overcoding".

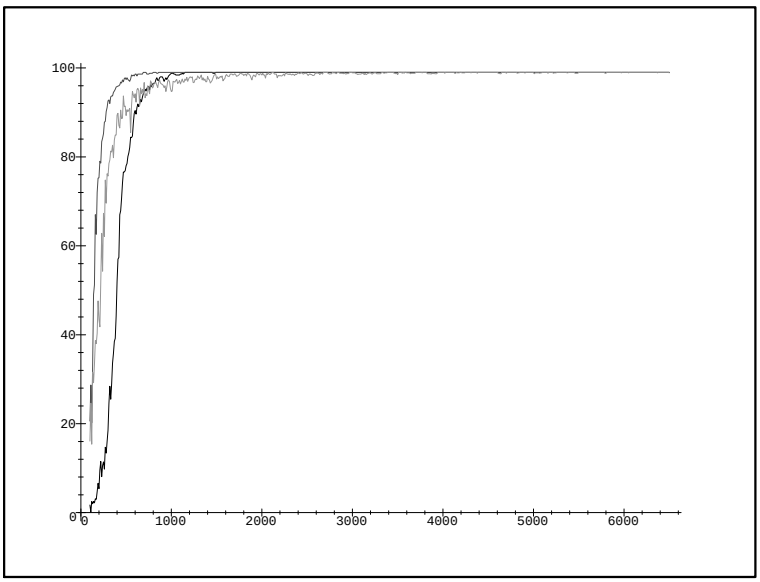


Figure 6: Experimental plots of hypothesis lengths $\tilde{d}_{\mathrm{MDL}}$ (most rapid initial increase), $\tilde{d}_{\mathrm{CV}}$ (intermediate initial increase) and $\tilde{d}_{\mathrm{GRM}}$ (least rapid initial increase) as a function of sample size for a target function of 100 alternating intervals and noise rate $\eta = 0.0$. Each data point represents an average over 10 trials. In this noise-free case, all three methods rapidly settle on the target length.

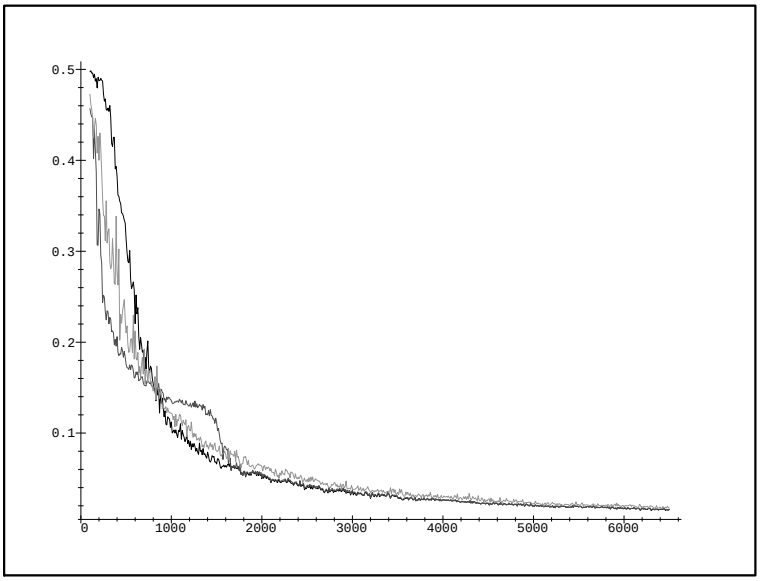


Figure 7: Experimental plots of generalization errors $\epsilon_{\mathrm{MDL}}$ (most rapid initial decrease), $\epsilon_{\mathrm{CV}}$ (intermediate initial decrease) and $\epsilon_{\mathrm{GRM}}$ (least rapid initial decrease) as a function of sample size for a target function of 100 alternating intervals and noise rate $\eta = 0.10$. Each data point represents an average over 10 trials. Note the appearance of a second regime in the relative behavior of MDL and GRM with the introduction of noise.

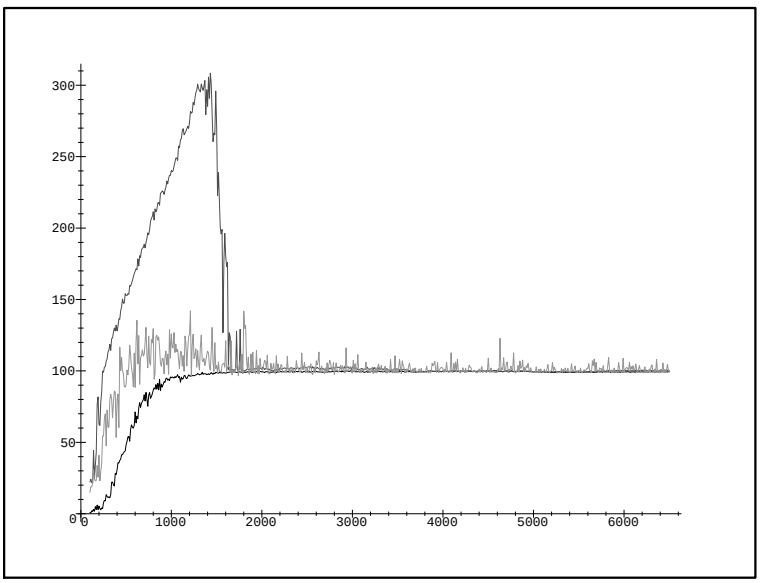


Figure 8: Experimental plots of hypothesis lengths $\tilde{d}_{\mathrm{MDL}}$ (most rapid initial increase), $\tilde{d}_{\mathrm{CV}}$ (intermediate initial increase) and $\tilde{d}_{\mathrm{GRM}}$ (least rapid initial increase) as a function of sample size for a target function of 100 alternating intervals and noise rate $\eta = 0.10$. Each data point represents an average over 10 trials. Note the correspondence between MDL's rapid decay in $\epsilon_{\mathrm{MDL}}$ shortly after $m = 2000$ and the rapid drop of $\tilde{d}_{\mathrm{MDL}}$ to the target value of 100.

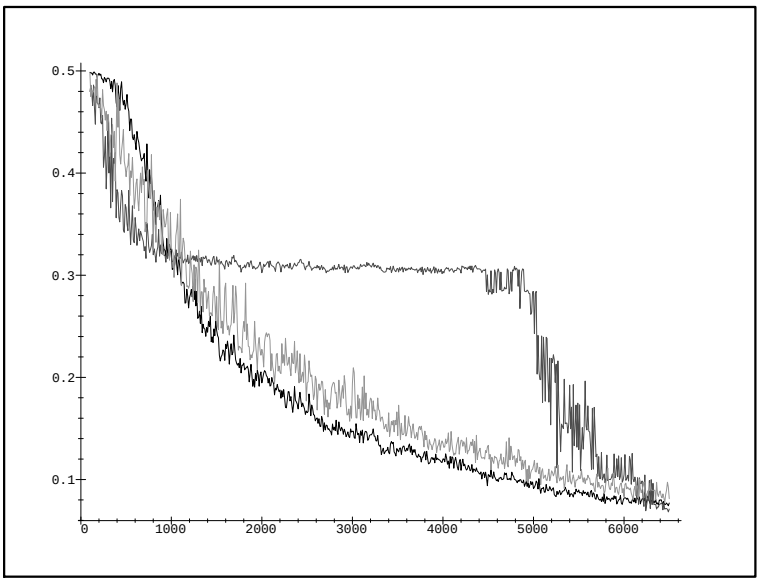


Figure 9: Experimental plots of generalization errors $\epsilon_{\mathrm{MDL}}$ (most rapid initial decrease), $\epsilon_{\mathrm{CV}}$ (intermediate initial decrease) and $\epsilon_{\mathrm{GRM}}$ (least rapid initial decrease) as a function of sample size for a target function of 100 alternating intervals and noise rate $\eta = 0.30$. Each data point represents an average over 10 trials. Notice the increasing variance of CV performance as the noise rate increases; this variance disappears asymptotically, but shows clearly at small sample sizes.

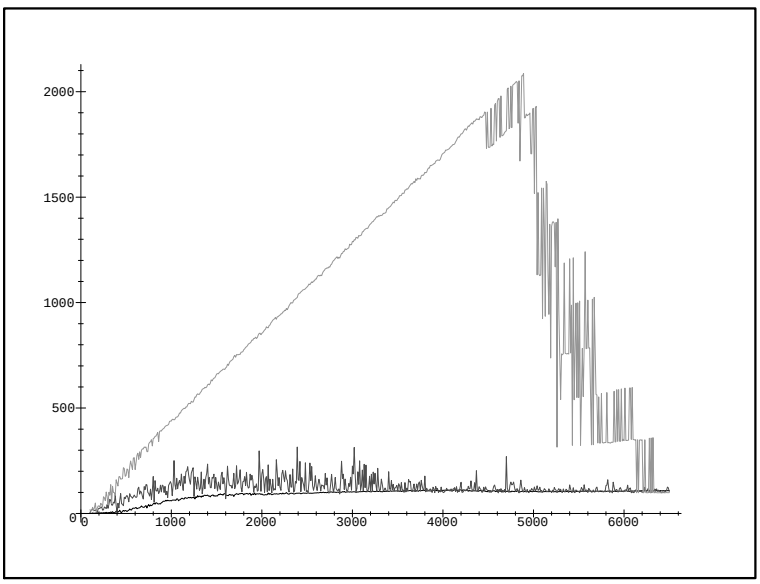


Figure 10: Experimental plots of hypothesis lengths $\tilde{d}_{\mathrm{MDL}}$ (most rapid initial increase), $\tilde{d}_{\mathrm{CV}}$ (intermediate initial increase) and $\tilde{d}_{\mathrm{GRM}}$ (least rapid initial increase) as a function of sample size for a target function of 100 alternating intervals and noise rate $\eta = 0.30$. Each data point represents an average over 10 trials. In this and the other plots, the apparent quantization of $\tilde{d}_{\mathrm{MDL}}$ during its transition down to the target value of 100 is an artifact of the averaging; on any given run, the method will choose between one of the two competing local minima at $d = 100$ and the point of consistency with the sample. The 11 quantized values for $\tilde{d}_{\mathrm{MDL}}$ observed during this transition simply represent the number of times $(0, \ldots, 10)$ that one of the minima can be visited out of 10 trials.

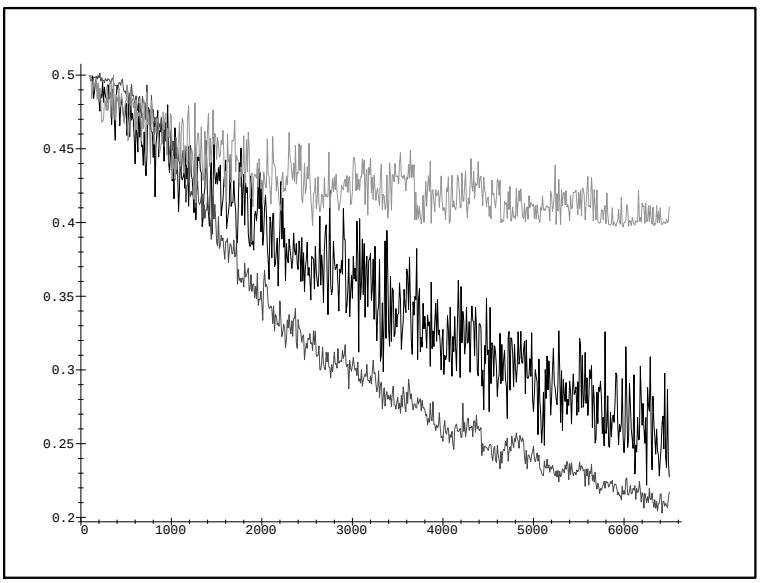


Figure 11: Experimental plots of generalization errors $\epsilon_{\mathrm{MDL}}$ (top plot), $\epsilon_{\mathrm{CV}}$ (intermediate plot) and $\epsilon_{\mathrm{GRM}}$ (bottom plot) as a function of sample size for a target function of 100 alternating intervals and noise rate $\eta = 0.40$. Each data point represents an average over 10 trials. At this large noise rate, $\epsilon_{\mathrm{MDL}}$ fails to transition from its shelf at $\eta$ even by $m = 15000$.

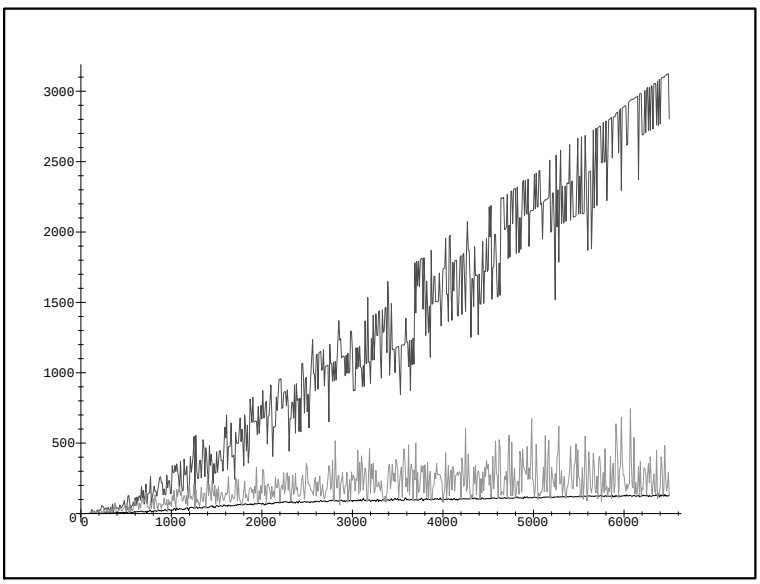


Figure 12: Experimental plots of hypothesis lengths $\tilde{d}_{\mathrm{MDL}}$ (top plot), $\tilde{d}_{\mathrm{CV}}$ (intermediate plot) and $\tilde{d}_{\mathrm{GRM}}$ (bottom plot) as a function of sample size for a target function of 100 alternating intervals and noise rate $\eta = 0.40$. Each data point represents an average over 10 trials.

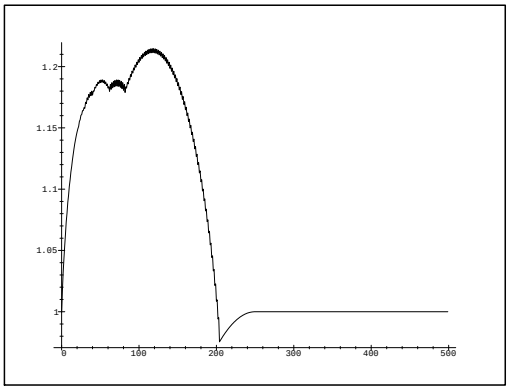


Figure 13: MDL penalty as a function of complexity $d$ for a single run on 500 examples of a target function of 100 alternating intervals and noise rate $\eta = 0.20$. Notice the appearance of a local minimum near the target length of 100.

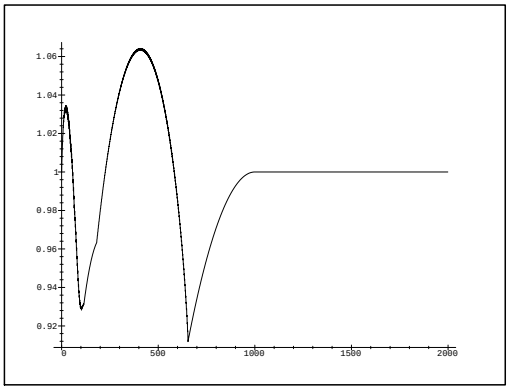


Figure 14: MDL total penalty $\mathcal{H}(\hat{\epsilon}(d)) + \mathcal{H}(d/m)$ versus complexity $d$ for a single run on 2000 examples of a target function of 100 alternating intervals and noise rate $\eta = 0.20$. There is a local minimum at approximately $d = 100$, and the global minimum at the point of consistency with the noisy sample.

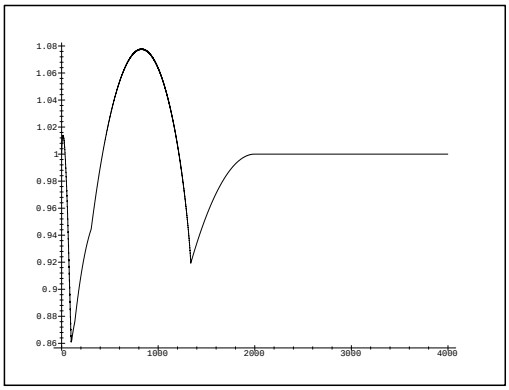


Figure 15: MDL total penalty $\mathcal{H}(\hat{\epsilon}(d)) + \mathcal{H}(d/m)$ versus complexity $d$ for a single run on 4000 examples of a target function of 100 alternating intervals and noise rate $\eta = 0.20$. The global minimum has now switched from the point of consistency to the target value of 100.

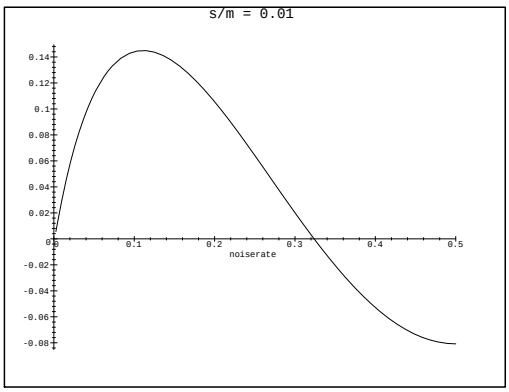


Figure 16: Plot of the function $\mathcal{H}(2\eta(1-\eta)+(s/m)(1-2\eta)^2)-\mathcal{H}(\eta)-\mathcal{H}(s/m)$ as a function of $\eta$ for $s/m = 0.01$. Positive values predict that MDL will choose the "correct" complexity $d = s$, while negative values predict that MDL will "overcode" by choosing $d = d_0$. For this value of $s/m$, increasing the noise rate can only cause degradation of performance. However, note the nonmonotonic behavior.

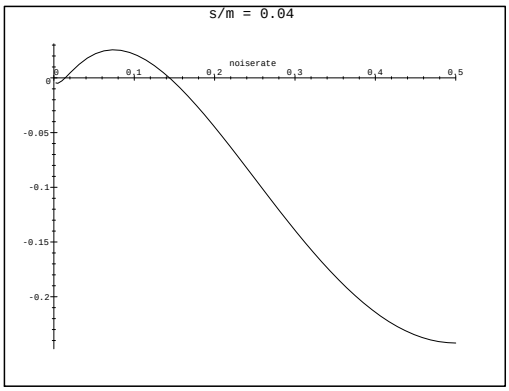


Figure 17: Plot of the function $\mathcal{H}(2\eta(1-\eta)+(s/m)(1-2\eta)^2)-\mathcal{H}(\eta)-\mathcal{H}(s/m)$ as a function of $\eta$ for $s/m = 0.04$. Note the behavior near 0, and see Figure 18.

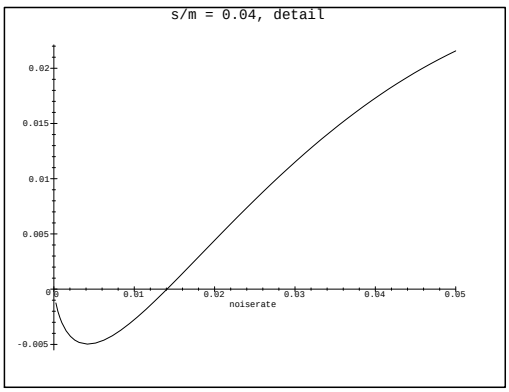


Figure 18: Detail of Figure 17 for small $\eta$. Here the nonmonotonic behavior has an interesting effect: increasing the noiserate may actually cause the value of $d$ chosen by MDL to move from $d = d_0$ to the superior $d = s$.

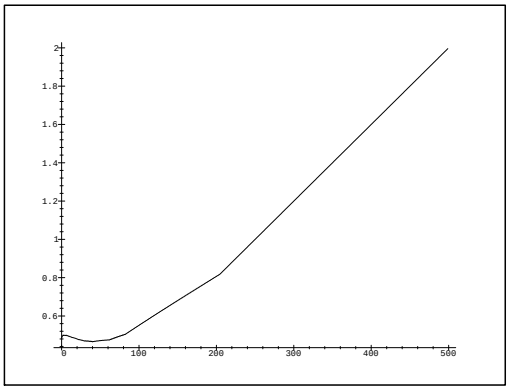


Figure 19: GRM penalty as a function of complexity $d$ for a single run on 500 examples of a target function of 100 alternating intervals and noise rate $\eta = 0.20$.

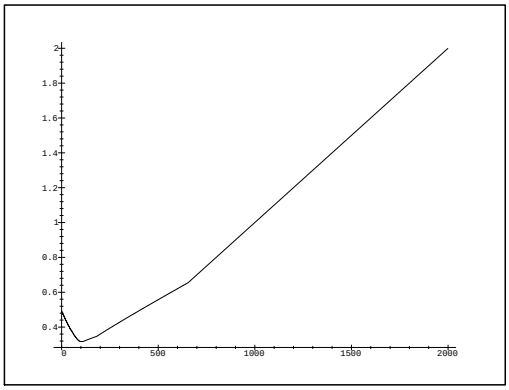


Figure 20: GRM total penalty $\hat{\epsilon}(d) + (d/m)(1 + \sqrt{1 + \hat{\epsilon}(d)m/d})$ versus complexity $d$ for a single run on 2000 examples of a target function of 100 alternating intervals and noise rate $\eta = 0.20$.

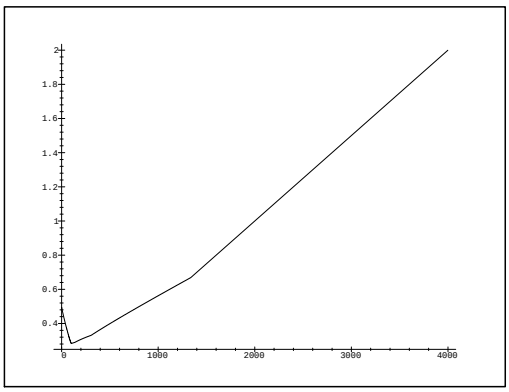


Figure 21: GRM penalty as a function of complexity $d$ for a single run on 4000 examples of a target function of 100 alternating intervals and noise rate $\eta = 0.20$.

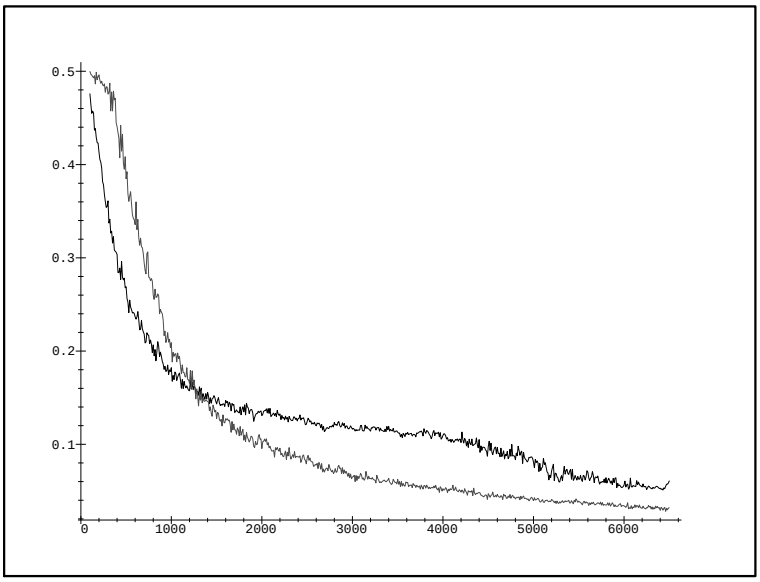


Figure 22: Experimental plots of generalization error $\epsilon_{\mathrm{GRM}}(m)$ using complexity penalty multipliers 1.0 (slow initial decrease) and 0.5 (rapid initial decrease) on the complexity penalty term $(d/m)(1+\sqrt{1+\hat{\epsilon}(d)m/d})$ versus sample size $m$ on a target of 100 alternating intervals and noise rate $\eta = 0.20$. Each data point represents an average over 10 trials.

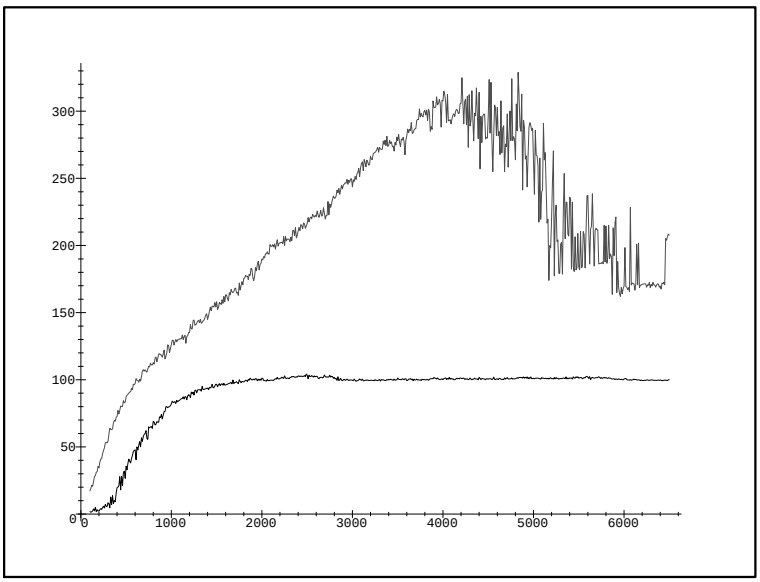


Figure 23: Experimental plots of hypothesis length $\tilde{d}_{\mathrm{GRM}}(m)$ using complexity penalty multipliers 1.0 (slow initial increase) and 0.5 (rapid initial increase) on the complexity penalty term $(d/m)(1+\sqrt{1+\hat{\epsilon}(d)m/d})$ versus sample size $m$ on a target of 100 alternating intervals and noise rate $\eta = 0.20$. Each data point represents an average over 10 trials.

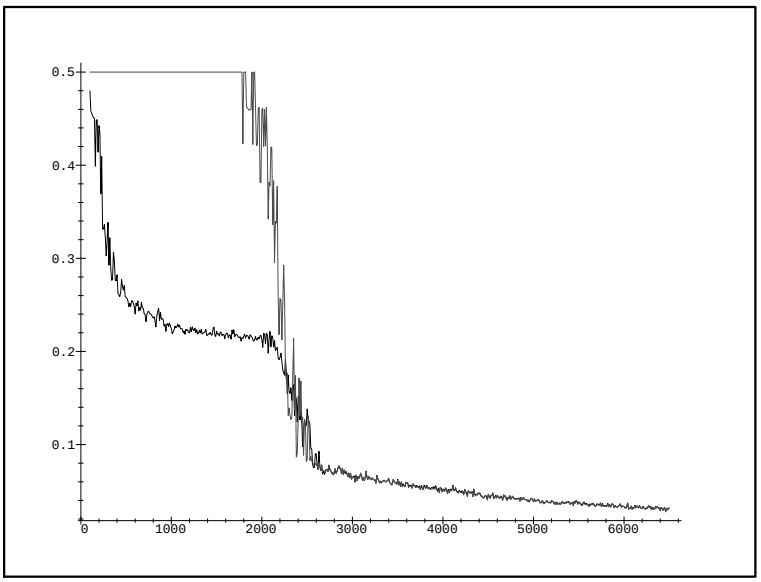


Figure 24: Experimental plots of generalization error $\epsilon_{\mathrm{MDL}}$ using complexity penalty multipliers 1.0 (rapid initial decrease) and 1.25 (slow initial decrease) as a function of sample size on a target of 100 alternating intervals and noise rate $\eta = 0.20$. Each data point represents an average over 10 trials.

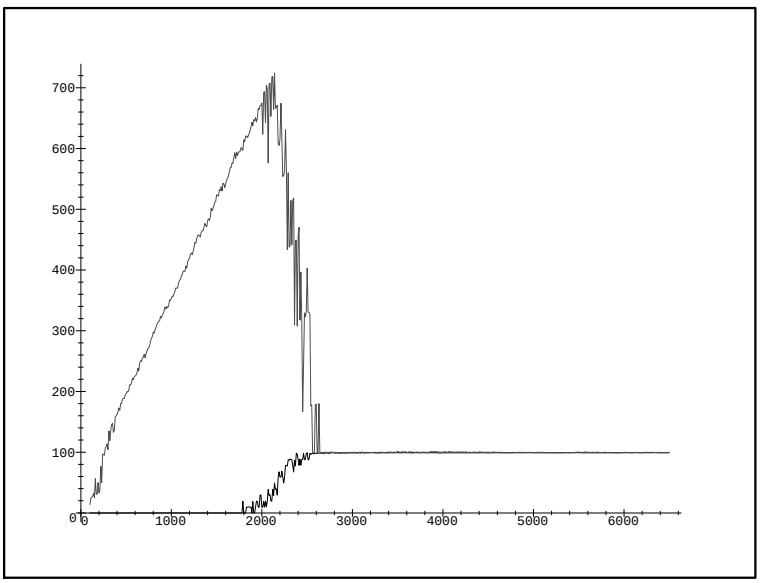


Figure 25: Experimental plots of hypothesis length $\tilde{d}_{\mathrm{MDL}}$ using complexity penalty multipliers 1.0 (rapid initial increase) and 1.25 (slow initial increase) as a function of sample size on a target of 100 alternating intervals and noise rate $\eta = 0.20$. Each data point represents an average over 10 trials. The adjustment to the rule here seems only to have caused damage, since the only effect is to keep $\tilde{d}_{\mathrm{GRM}}$ at 0 (undercoding) until $m$ is close to 2000, and then to rapidly approach 100 from below, whereas in the unmodified (constant penalty multiplier 1.0) rule $\tilde{d}_{\mathrm{GRM}}$ approached 100 from above at approximately the sample sample size, but achieved nontrivial generalization error in the initial overcoding region. Some simple calculations indicate that even if the constant is increased only to the value 1.0000001, the approach to 100 from below will still not commence until $m > 2000$. Larger values for the constant will of course only perform even more poorly.

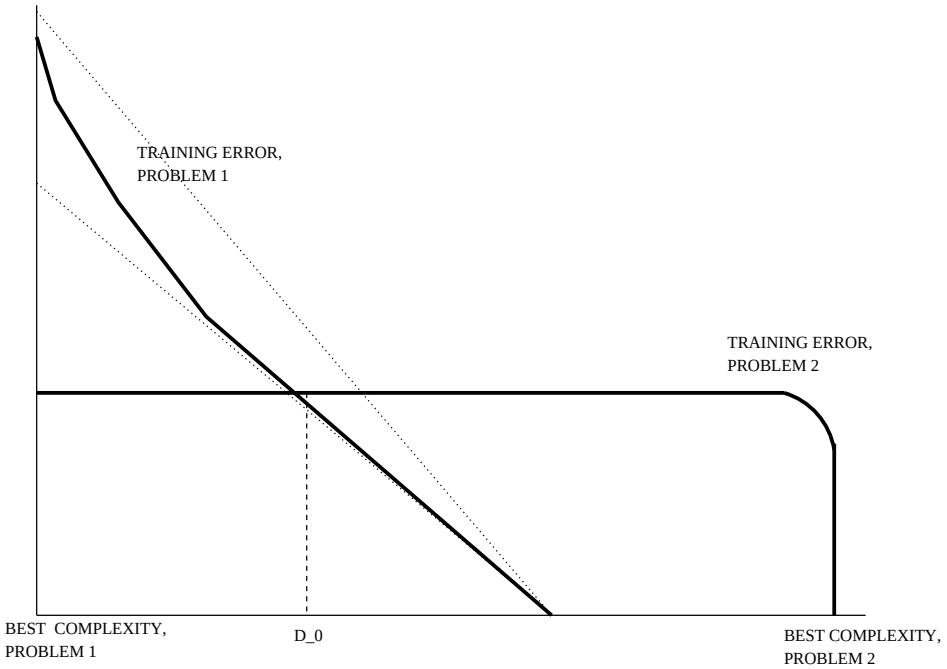


Figure 26: Figure illustrating the proof of Theorem 6. The dark lines indicate typical behavior for the two training error curves $\hat{\epsilon}_1(d)$ and $\hat{\epsilon}_2(d)$, and the dashed lines indicate the provable bounds on $\hat{\epsilon}_1(d)$. For legibility, we use the notation D_0 to indicate the intersection point $d_0$ of the proof.

## Appendix: Experimental Details

All experimental results described in this paper are obtained for the intervals model selection problem. Recall that in this problem, the function class $F_d$ consists of all boolean functions over the domain $[0,1]$ which have at most $d$ alternations of label. There are two main reasons for choosing this problem for our investigation. The first is that the complexity of the hypothesis functions is unlimited; in particular, it is not hard to show that the Vapnik-Chervonenkis dimension of $F_d$ is $d$, and thus as $d$ increases we allow arbitrarily complex functions. The second reason is that this is one of the few cases in which training error minimization is feasible[11]. (A number of papers provide evidence for the intractability of training error minimization for a variety of natural function classes [8, 2, 7].)

More precisely, there is an algorithm that on input an *arbitrary* sample $S = \{\langle x_i, b_i \rangle\}$ (where $x_i \in [0,1]$ and $b_i \in \{0,1\}$) and complexity value $d$, outputs a function in $VS_S(d)$. The algorithm is based on dynamic programming, and a straightforward implementation yields a running time that is $O(dm^2)$. However, we have developed a more sophisticated implementation that yields a running time that is $O(m \log m)$. The algorithm was implemented in the C++ programming language on an SGI Challenge XL with 8 150 MHz processors and 1 gigabyte of RAM. This implementation allowed execution of the training error minimization algorithm on samples of size up to $m \approx 15000$ in only a few seconds of real time.

The fast training error minimization code was the heart of a more elaborate experimental tool that offered the following features:

- The user specifies a target intervals function over $[0,1]$ in a file that indicates the values at which the function changes label. Thus, a file containing the values 0.15, 0.40, 0.75 specifies the boolean function that is 1 on the interval $[0, 0.15)$, 0 on the region $[0.15, 0.40)$, 1 on the region $[0.40, 0.75)$ and 0 on the region $[0.75, 1.0]$.

- The user specifies the sample size $m$, and the noise rate $\eta$ with which the labels in the sample will be corrupted with noise. The user also specifies one or more model selection algorithms, such as GRM, MDL or CV.

- A random sample $S$ of size $m$ of the specified target function corrupted by the specified noise rate is then generated by the program (inputs are drawn according to the uniform distribution). For each value of $d$ from 0 to $m$, $S$ and $d$ are then given to the training error minimization code. This code returns a function $\tilde{h}_d \in VS_S(d)$. If $VS_S(d)$ contains functions giving different labelings to $S$, the code chooses the least in a lexicographic ordering. The hypothesis selected from $VS_S(d)$ always has its label alternation points exactly midway between sample inputs.

- For each $\tilde{h}_d$, the true generalization error $\epsilon(\tilde{h}_d)$ is computed with respect to the specified target function, thus allowing exact computation of the curve $\epsilon(d)$.

- For each $\tilde{h}_d$, the total penalty assigned to $\tilde{h}_d$ by the chosen model selection algorithm is computed from $\tilde{h}_d$, $S$ and $d$. Minimization of this total penalty with respect to $d$ is then performed by the code, resulting in the hypothesis $\tilde{h}_{\tilde{d}}$ chosen by the specified model selection algorithm. The error of this hypothesis can then be compared with that of other model selection algorithms, as well as the optimal value $\min_d\{\epsilon(d)\}$.

[11] This is important in light of our earlier assertion that a good model selection algorithm should at least perform well when the underlying learning algorithm implements training error minimization, and we do not wish any of our experimental results to be artifacts of the unknown properties of heuristics such as backpropagation or ID3.

The experiments given in the paper were all performed using a target function of 100 regions of equal width and alternating label. The code provides an option for repeated trials at each sample size, which was used extensively in the experiments. The code produces plot files that were averaged where appropriate. The postscript plots shown were generated by reading the plot files generated by the code into the Xmaple system, which allows postscript output.