Average-Case Analysis of Classification Algorithms for Boolean Functions and Decision Trees[⋆]

Tobias Scheffer

University of Magdeburg, FIN/IWS
P.O. Box 4120, 39016 Magdeburg, Germany
scheffer@iws.cs.uni-magdeburg.de

Abstract. We conduct an *average-case analysis* of the generalization error rate of classification algorithms with finite model classes. Unlike *worst-case* approaches, we do not rely on bounds that hold for all possible learning problems. Instead, we study the behavior of a learning algorithm *for a given problem*, taking properties of the problem and the learner into account. The solution depends only on known quantities (*e.g.,* the sample size), and the histogram of error rates in the model class which we determine for the case that the sought target is a randomly drawn Boolean function. We then discuss how the error histogram can be estimated from a given sample and thus show how the analysis can be applied approximately in the more realistic scenario that the target is unknown. Experiments show that our analysis can predict the behavior of decision tree algorithms fairly accurately even if the error histogram is estimated from a sample.

1 Introduction

In the setting of *classification learning* which we study in this paper, the task of a *learner* is to approximate a joint distribution on *instances* and *class labels* as well as possible. A *hypothesis* is a mapping from instances to class labels; the (generalization, or true) *error rate* of a hypothesis h is the chance of drawing a pair of an instance x and a class label y (when drawing according to the sought target distribution) such that the hypothesis conjectures a class label $h(x)$ which is distinct from the "correct" class label y. While we would like to minimize this true error rate, it is only the empirical error on the training sample (*i.e.,* a set of pairs (x_i, y_i) of fixed size) which we can measure and thus minimize. A learner minimizes the empirical error within a prescribed model class (a set of potentially available hypotheses).

Most known *analyses* of classification algorithms give *worst-case* guarantees on the behavior of the studied algorithms. Typically, it is guaranteed that the performance of the learner is very unlikely to lie below some bound *for every*

[⋆] *Proceedings of the Eleventh International Conference on Algorithmic Learning Theory (ALT-2000).* Sydney, 2000. © Springer-Verlag.

possible underlying problem. Consequently, such bounds tend to be pessimistic for all but very few underlying learning problems.

In an attempt to close the gap between worst-case guarantees and experimental results, a number of *average-case* analyses have been presented which predict the expected behavior (over all possible samples) of a learning algorithm *for a given problem.* Average-case analyses have been presented for decision stump learners [7], k-nearest neighbor [11,12], and linear neural networks [3] as well as for one-variable pattern languages [13] and naive Bayesian classifiers [10,9].

PAC- and VC-style results impose mathematical constraints on the range of possible error rates of classification algorithms which hold for all possible learning problems. Complementing this *mathematical* view, average-case analyses can be seen as reflecting a *science*-oriented perspective. The learning agent is considered as a system the behavior of which is to be described as accurately as possible. The primary benefit of average-case analyses is their ability to predict the behavior of a learning algorithm in a specific scenario much better than worst-case analyses; their primary drawback is their dependence on properties of the learning algorithm and the learning problem which correspond to the the initial state of the system. In a typical classification setting, these properties are unknown.

In Sections 2 and 3, we present computationally efficient *average-case* analyses that predict the behavior of classification algorithms with finite hypothesis languages. In Section 2 we assume that the training set error of the returned hypothesis is known and quantify the expected generalization error of hypotheses with that empirical error. In Section 3 we assume that the learner finds the training set error minimizing hypothesis in the model class (but this least training set error does not have to be known) and quantify the expected generalization error of that hypothesis. Both analyses depend on the histogram of error rates in the model class. This joint property of model class and learning problem counts how often each possible error rate occurs in the model class.

In Section 4, we derive the exact error histogram for the case that the sought target is a randomly drawn function and the instances are governed by the uniform distribution. Similar settings are commonly studied in average-case analyses (*e.g.,* [7]). In Section 5, we discuss how the error histogram can be estimated from an available sample. We can then apply the analysis approximately for arbitrary targets. We present experiments that indicate that, even without any background knowledge on the target, we can still obtain fairly accurate results.

Let us clarify some notational details. Let H_i be some finite model class – *i.e.,* a set of available hypotheses. For instance, H_i could contain all decision trees with i leaf nodes. $h \in H_i$ is then a hypothesis and maps instances x to class labels y. A classification problem is given by an (unknown) density $p(x,y)$. The generalization error rate of h with respect to this problem (which we want to minimize) is then $\epsilon(h) = \int \sum_y \ell(h(x),y)p(x,y)dx$, where $\ell(\cdot,\cdot)$ is the zero-one loss function. Given a finite sample S consisting of m independent examples, drawn according to $p(x,y)$, the empirical (or sample) error rate of h is $e(h) = \frac{1}{m}\sum_{(x,y)\in S} \ell(h(x),y)$. It is important to distinguish between generalization error

ϵ (which we really *want* to minimize) and empirical error e (which we *are able* to measure and minimize using the sample) throughout this paper.

2 Generalization Error Given the Empirical Error

Suppose that we have a given model class H_i and a sample size m. The model class H_i is the particular learning bias of the learning algorithm, the behavior of which we would like to predict. Every hypothesis $h \in H_i$ has a fixed but unknown generalization error $\epsilon(h)$ with respect to the (unknown) learning problem $p(x, y)$. When we draw a sample S governed by $p(x, y)^m$, then each hypothesis incurs an empirical error rate $e(h)$. Suppose that we put the hypotheses into boxes labeled with the possible empirical error rates $\frac{0}{m}, \frac{1}{m}, \ldots, \frac{m}{m}$. We call the set of hypotheses in box e H_i^e. Each box with label e has its own distribution of generalization error rates in it (over all possible samples S and over the hypotheses contained in the box). We will write this distribution $p(\epsilon(h_i^e)|e, H_i, m)$. We would expect most of the hypotheses with empirical error rate of $\frac{0}{m}$ to have fairly small generalization error rates, although the majority of them is likely to incur a nonzero generalization error. On the other hand, most hypotheses with empirical error rate $\frac{m}{m}$ will also incur a rather high true error (depending on the sample size and other factors) which will in most cases still be lower than one.

A learning algorithm conducts a search in the prescribed model class H_i and comes to some hypothesis h_i^L with empirical error e (not necessarily the globally smallest empirical error in H_i). If we assume that all hypotheses in H_i with identical empirical error e are equally likely to be found by the learner, then h_i^L can be treated as if it were drawn from H_i^e (the box of hypotheses with empirical error e) under uniform distribution. Consequently, $p(\epsilon(h_i^e)|e, H_i, m)$ governs the generalization error of our learning algorithm when the observed empirical error of the returned hypothesis is e. When we can quantify $p(\epsilon(h_i^e)|e, H_i, m)$, then we can also quantify the distribution which governs the generalization error of the hypothesis returned by our learner.

We can read $p(\epsilon(h_i^e)|e, H_i, m)$ as "P(generalization error | empirical error)". The intuition of our analysis (which is a simplified version of the analysis discussed in [15]) is that application of Bayes' rule implies "P(generalization error | empirical error) = P(empirical error | generalization error)P(generalization error)/ normalization constant". Note that P(empirical error | generalization error) is simply the binomial distribution. (Each example can be classified correctly or erroneously; the chance of the latter happening is ϵ; this leads to a binomial distribution.) We can interpret "P(generalization error)", the prior in our equation, as the histogram of error rates in H_i. This histogram counts, for every ϵ the fraction of the hypotheses in H_i which incur an error rate of ϵ. Let us now look at the analysis in more detail.

Let h_i^L be a hypothesis drawn from H_i^e at random under uniform distribution. In Equation 1, we only expand our definition of h_i^L. Then, in Equation 2, we decompose the expectation by integrating over all possible error rates ϵ. In Equation 3, we apply Bayes' rule. $\pi(\epsilon|H_i)$ is the histogram of error rates in H_i. It specifies the probability of drawing a hypothesis with error rate ϵ when drawing at random under uniform distribution from H_i.

$$
\begin{aligned}
&E(\epsilon(h_i^L)|e, H_i, m) \\
&= E(\epsilon(h)|e(h) = e, h \in H_i, m) && (1) \\
&= \int \epsilon p(\epsilon(h) = \epsilon | e(h) = e, h \in H_i, m) d\epsilon && (2) \\
&= \int \epsilon \frac{P(e(h) = e|\epsilon(h) = \epsilon, h \in H_i, m)\pi(\epsilon|H_i)}{P(e(h) = e|h \in H_i, m)} d\epsilon && (3)
\end{aligned}
$$

Since, over all ϵ, the distribution $p(\epsilon(h) = \epsilon|e(h) = e, H_i, m)$ has to integrate to one (Equation 4), we can treat $P(e(h) = e|h \in H_i, m)$ as a normalizing constant which we can determine as in Equation 6.

$$
\begin{aligned}
&\int p(\epsilon(h) = \epsilon|e(h) = e, h \in H_i, m) d\epsilon = 1 && (4) \\
&\Leftrightarrow \int \frac{P(e(h) = e|\epsilon(h) = \epsilon, h \in H_i, m)\pi(\epsilon|H_i)}{P(e(h) = e|h \in H_i, m)} d\epsilon = 1 && (5) \\
&\Leftrightarrow P(e(h) = e|h \in H_i, m) = \int P(e(h) = e|\epsilon(h) = \epsilon, h \in H_i, m)\pi(\epsilon|H_i) d\epsilon && (6)
\end{aligned}
$$

Combining Equations 3 and 6 we obtain Equation 7. In this equation, we also state that, when the true error ϵ is given, the empirical error e is governed by the binomial distribution which we write as $B[\epsilon, m](e)$.

$$
E(\epsilon(h_i^L)|e, H_i, m) = \frac{\int \epsilon B[\epsilon, m](e)\pi(\epsilon|H_i) d\epsilon}{\int B[\epsilon, m](e)\pi(\epsilon|H_i) d\epsilon} \qquad (7)
$$

We have now found a solution that quantifies $E(\epsilon(h_i^L)|e, H_i, m)$, *the exact* expected generalization error of a hypothesis from H_i with empirical error rate e for a given learning problem $p(x, y)$. Equation 7 specifies the actual error rate for the given learning problem rather than a worst-case bound that holds for all possible learning problems. The additional information of $\pi(\epsilon|H_i)$ makes this possible.

3 Analysis of Exhaustive Learners

In this section, we assume that the learner can be guaranteed to find the hypothesis in H_i that minimizes the empirical error (breaking ties by drawing at random). On the other hand, we do not require the empirical error rate of the resulting hypothesis to be known (so the learner does not have to be invoked before the analysis can be applied). We can predict both the resulting empirical error rate and the resulting generalization error from the histogram of error rates and the number of hypotheses. The analysis is a simplification of an analysis proposed by Scheffer and Joachims [19]. Let us first sketch how the resulting empirical error rate on the training set can be predicted without running the learning algorithm at all.

The empirical error rate of a single hypothesis with generalization error ϵ is governed by the binomial distribution $B[m, \epsilon]$. The least empirical error rate in H_i is e if no hypothesis achieves an empirical error which is lower than e. Let us make the simplifying assumption that the empirical error rates of two or more hypotheses are independent *given the corresponding true error rates.* Formally, $P(\bigwedge_{h_j \in H_i} e(h_j)|\epsilon(h_j)) = \prod_{h_j \in H_i} P(e(h_j)|\epsilon(h_j))$. Now we can approximate the chance that no hypothesis incurs an error of less than e as $\prod_{h \in H_i} P(e(h) \geq e|\epsilon(h), m)$. Note that the histogram $\pi(\epsilon|H_i)$ tells us how many hypotheses have error rates of ϵ (for each ϵ). Let us now look at the analysis in more detail.

In order to determine the expected true error (expected over all samples) of h_i^L (the hypothesis that minimizes the empirical error within H_i), we factorize the hypothesis h that the learner returns (Equation 8). Since we assume the learner to break ties between hypotheses with equally small empirical error at random, all hypotheses with equal true error rates ϵ have an exactly equal prior probability of becoming h_i^L. We re-arrange Equation 8 such that all hypotheses h_ϵ with true error ϵ are grouped together. $\pi(\epsilon|H_i)$ is again the density of hypotheses with error rate ϵ among all the hypotheses in H_i (with respect to the given learning problem). This takes us to Equation 9.

$$\begin{aligned} E(\epsilon(h_i^L)|H_i, m) &= \int_h \epsilon(h) P(h_i^L = h|H_i, m) dh & (8) \\ &= \int_\epsilon \epsilon P(h_i^L = h_\epsilon|\epsilon, H_i, m) \pi(\epsilon|H_i) d\epsilon & (9) \end{aligned}$$

Let $H_i^* = \text{argmin}_{h \in H_i}\{e(h)\}$ be the set of hypotheses in H_i which incur the least empirical error rate. Note that H_i^* is a random variable because only the sample size m is fixed whereas the sample S itself (on which H_i^* depends) is a random variable. In order to determine the chance that h_ϵ (an arbitrary hypothesis with true error rate ϵ) is selected as h_i^L, we first factorize the chance that h_ϵ lies in H_i^*, the empirical error minimizing hypotheses of H_i (Equation 10). A hypothesis that does not lie in H_i^* has a zero probability of becoming h_i^L (Equation 11). In Equation 12, we factorize the cardinality of $|H_i^*|$. When this set is of size n, then each hypothesis in H_i^* has a chance of $\frac{1}{n}$ of becoming h_i^L (the learner breaks ties at random) (Equation 13). In Equation 14, we factorize the least empirical error e and, in Equation 15, we simply split up the conjuction (like $p(a, b) = p(a)p(b|a)$).

$$\begin{aligned} & P(h_i^L = h_\epsilon|\epsilon, H_i, m) \\ & = P(h_i^L = h_\epsilon|H_i, m, h_\epsilon \in H_i^*) P(h_\epsilon \in H_i^*) & (10) \\ & \quad + P(h_i^L = h_\epsilon|H_i, m, h_\epsilon \notin H_i^*)(1 - P(h_\epsilon \in H_i^*)) \\ & = P(h_i^L = h_\epsilon|H_i, m, h_\epsilon \in H_i^*) P(h_\epsilon \in H_i^*) & (11) \end{aligned}$$

$$= \sum_n P\big(h_i^L = h_\epsilon \big| H_i, m, h_\epsilon \in H_i^*, |H_i^*| = n\big) P(h_\epsilon \in H_i^*, |H_i^*| = n) \tag{12}$$

$$= \sum_n \frac{1}{n} P(h_\epsilon \in H_i^*, |H_i^*| = n) \tag{13}$$

$$= \sum_e \sum_n \frac{1}{n} P\big(h_\epsilon \in H_i^*, |H_i^*| = n \big| e(h_\epsilon) = e\big) P(e(h_\epsilon) = e|\epsilon, m) \tag{14}$$

$$\begin{aligned}= \sum_e \sum_n \frac{1}{n} P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) P\big(|H_i^*| = n \big| h_\epsilon \in H_i^*, e(h_\epsilon) = e\big) \\ P(e(h_\epsilon) = e|\epsilon, m)\end{aligned} \tag{15}$$

By inserting Equation 15 into Equation 9 we get Equation 16.

$$\begin{aligned}&E(\epsilon(h_i^L)|H_i, m) \\ &= \int \epsilon \Bigg(\sum_e \sum_n \frac{1}{n} P\big(|H_i^*| = n \big| h_\epsilon \in H_i^*, e(h_\epsilon) = e\big) \\ &\qquad P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) P(e(h_\epsilon) = e|\epsilon, m) \pi(\epsilon|H_i) \Bigg) d\epsilon\end{aligned} \tag{16}$$

Assuming that the chance of the set of empirical error minimizing hypotheses H_i^* being of size n when h_ϵ is known to lie in this set does not depend on *which* hypothesis is known to lie in this set (formally, $P\big(|H_i^*| = n \big| h_1 \in H_i^*\big) = P\big(|H_i^*| = n \big| h_2 \in H_i^*\big)$ for all h_1, h_2) we can claim that $c = P\big(|H_i^*| = n \big| h_\epsilon \in H_i^*, e(h_\epsilon) = e\big)$ is constant for all h_ϵ.

Equation 16 specifies the expectation of $\epsilon(h_i^L)$. The density $p(\epsilon(h_i^L)|H_i, m)$ has to integrate to one (Equation 17). Equation 16 takes us from Equation 17 to Equation 18 in which we use the abbreviation c for $P\big(|H_i^*| = n \big| h_\epsilon \in H_i^*, e(h_\epsilon) = e\big)$. c is therefore determined uniquely by Equation 19.

$$\int p(\epsilon(h_i^L) = \epsilon | H_i, m) d\epsilon = 1 \tag{17}$$

$$\begin{aligned}\Leftrightarrow \int \sum_e \sum_n \frac{1}{n} c \; P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) \\ P(e(h_\epsilon) = e|\epsilon, m) \pi(\epsilon|H_i) d\epsilon = 1\end{aligned} \tag{18}$$

$$\Leftrightarrow c = \left(\int \sum_e P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) P(e(h_\epsilon) = e|\epsilon, m) \pi(\epsilon|H_i) d\epsilon \right)^{-1} \tag{19}$$

Combining Equations 16 and 19 and stating that the empirical error is governed by the binomial distribution (given the true error) we obtain Equation 20.

$$\begin{aligned}&E(\epsilon(h_i^L)|H_i, m) \\ &= \frac{\int \epsilon \left(\sum_e P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) B[\epsilon, m](e) \pi(\epsilon|H_i) \right) d\epsilon}{\int \sum_e P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) B[\epsilon, m](e) \pi(\epsilon|H_i) d\epsilon}\end{aligned} \tag{20}$$

Let us now tackle the last unknown term, $P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m)$. A hypothesis h_ϵ (with true error rate ϵ) lies in H_i^* when no hypothesis in H_i achieves a lower empirical error rate. There are $|H_i|$ many hypotheses; their true error rates are fixed but completely arbitrary – *i.e.,* they are neither independent nor governed by some identical distribution. These $|H_i|$ error rates constitute the density $\pi(\epsilon|H_i)$ which measures how often each error rate ϵ occurs in H_i (we have already seen this density in Equation 9). Each of these hypotheses incurs an empirical error rate that is by itself governed by the binomial distribution $B[m, \epsilon]$. Let us assume that the empirical error rates of two or more hypotheses are independent *given the corresponding true error rates* as discussed earlier in this section. Formally, $P(\bigwedge_{h_j \in H_i} e(h_j) | \epsilon(h_j)) = \prod_{h_j \in H_i} P(e(h_j)|\epsilon(h_j))$. Now we can quantify the chance that no hypothesis incurs an error of less than e which makes our hypothesis h with $e(h) = e$ a member of H_i^*. For all but extremely small H_i (formally, $p^{|H_i|} \approx p^{|H_i|-1}$) we can write this chance as in Equation 21. Note again that the empirical error (given the true error) is governed by the binomial distribution (Equation 22).

$$P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) = \prod_{\epsilon'} P(e(h) \geq e | \epsilon', m)^{|H_i| \pi(\epsilon'|H_i)} \tag{21}$$

$$= \prod_{\epsilon'} \left(\sum_{e' \geq e} B[\epsilon', m](e') \right)^{|H_i| \pi(\epsilon'|H_i)} \tag{22}$$

What have we achieved so far? Equations 20 and 22 quantify the expected generalization error of h_i^L for a given problem in terms of three quantities: the number of hypotheses in model class H_i (which can typically easily be computed), the sample size m (which is known), and the histogram of error rates in H_i, $\pi(\epsilon|H_i)$. Note that, for Equations 20 to give us the expected error $\epsilon(h_i^L)$, it is not necessary to actually run the learner and determine $e(h_i^L)$. Let us also emphasize that we are not talking about *bounds* on the error rate for a class of possible problems. Subject to the mentioned independence assumptions, Equations 20 and 21 quantify *the* expected generalization error of an empirical error minimizing hypothesis *for a particular, given learning problem.* When only the sample size m and $|H_i|$ are given, it is impossible to determine where in the interval specified by the Chernoff bound the actual error rate lies. Additionally given the density $\pi(\epsilon|H_i)$, however, we can determine the *actual* density that governs the generalization error, and thereby also the expected generalization error.

4 Learning Boolean Functions

In order to apply the analysis, the histogram of error rates $\pi(\epsilon|H_i)$ has to be known. Let us determine $\pi(\epsilon|H_i)$ when the target is a randomly drawn Boolean function over attributes x_1 through x_k and the instances are governed by the uniform distribution. For each target function f_k the target distribution $p_k(x, y)$

is then $\frac{1}{|X|}$ when $f_k(x) = y$ and 0 otherwise. Let H_i contain all Boolean functions over the first i attributes. Model classes H_1 to H_{k-1} contain 1 through $k-1$ of the relevant attributes; the target function does usually not lie within the model class and the classifier can only approximate the target. Model class H_k contains all relevant attributes. Model classes H_{k+1} through H_n contain all relevant plus additional irrelevant attributes.

Each target function f_k (with corresponding target distribution $p_k(x, y)$) yields some error histogram $\pi_k(\epsilon|H_i, f_k) = \pi_k(\epsilon|H_i, p_k(x, y))$. When $P(f_k)$ is the uniform distribution as stated above, then the expected resulting error can be described by Equation 23, which is just Equation 20 averaged over all f_k. The subscript $E_{\{f_k,S\}}$ indicates that now both f_k and the sample S are random variables.

$$\begin{aligned} & E_{\{f_k,S\}}(\epsilon(h_L)|H_i, m) && (23) \\ = \ & \int \frac{\int \epsilon \left(\sum_e P(h_\epsilon \in H_i^*|e(h_\epsilon) = e, m) B[\epsilon, m](e) \pi(\epsilon|H_i, f_k)\right) d\epsilon}{\int \sum_e P(h_\epsilon \in H_i^*|e(h_\epsilon) = e, m) B[\epsilon, m](e) \pi(\epsilon|H_i, f_k) d\epsilon} P(f_k) df_k \end{aligned}$$

$$\text{where } P(h_\epsilon \in H_i^*|e(h_\epsilon) = e, m) = \prod_{\epsilon'} \left(\sum_{e' \geq e} B[\epsilon', m](e') \right)^{|H_i|\pi(\epsilon'|H_i)} \quad (24)$$

In order to further reduce Equation 23 we need to distinguish two cases.

(1): $i < k$. f_k splits the Boolean instance space into 2^k instances whereas the hypotheses split the space only into 2^i subspaces each of which is assigned only one class label. Hence, 2^{k-i} instances with potentially distinct class labels fall into the same subspace. Since f_k is governed by the uniform distribution, assigning one class label (drawn uniformly from the set $\{0, 1\}$ to 2^{k-i} instances will misclassify a number ν of instances governed by the binomial distribution $B[2^{k-i}, \frac{1}{2}]$. Let ν_1 through ν_{2^i} be the numbers of instances misclassified in subspaces 1 through 2^i when a randomly drawn class label is assigned to the whole subspace. The vector $(\nu_1, \ldots, \nu_{2^i})$ is governed by $(B[2^{k-i}, \frac{1}{2}])^{2^i}$ as specified more detailedly in Equation 25.

$$P(\nu = (\nu_1, \ldots, \nu_{2^i})) = \prod_{j=1}^{j=2^i} B\left[2^{k-i}, \frac{1}{2}\right](\nu_j) \quad (25)$$

Given a vector ν, the corresponding error rate is just the sum over all subspaces divided by the number of instances: $\epsilon = \sum_{j=1}^{2^i} \nu_j / 2^k$. Hence, we can characterize the distribution that governs this sum of errors ϵ recursively in Equation 26. The intuition of this equation is that an error of e instances is incurred in subspace j through 2^i when *either* an error of ν_j (class label 0) is incurred in subspace j and an error of $e - \nu_j$ is incurred in subspaces $j + 1$ through k, *or* an error of $2^{k-i} - \nu_j$ is incurred in subspace j (class label 1) and the remaining error of $e - (2^{k-i} - \nu_j)$ is incurred in subspaces $j + 1$ through k. The factor 2^k is used to convert error rates into absolute numbers of errors and vice versa. The intuition of Equation 27 is that, in the last subspace, ν_{2^i} instances are misclassified with certainty when $\nu_{2^i} = 2^{k-i} - \nu_{2^i}$ (equally many instances have class labels of zero and one), and ν_{2^i} and $2^{k-i} - \nu_{2^i}$ instances are misclassified with probability $\frac{1}{2}$ otherwise, and no other error rates are possible.

$$P\left(\epsilon = \frac{e}{2^k}\middle|\nu_j, \ldots, \nu_{2^i}\right) = \frac{1}{2}P\left(\epsilon = \frac{e-\nu_1}{2^k}\middle|\nu_{j+1}, \ldots, \nu_{2^i}\right) + \frac{1}{2}P\left(\epsilon = \frac{e - 2^{k-i} + \nu_1}{2^k}\middle|\nu_{j+1}, \ldots, \nu_{2^i}\right) \quad (26)$$

$$\text{where } P\left(\epsilon = \frac{e}{2^k}\middle|\nu_{2^i}\right) = \begin{cases} 1 \text{ iff } \nu_{2^i} = 2^{k-i} - \nu_{2^i} \\ \frac{1}{2} \text{ iff } \nu_{2^i} = e \\ \frac{1}{2} \text{ iff } \nu_{2^i} = 2^{k-i} - e \\ 0 \text{ otherwise} \end{cases} \quad (27)$$

Hence, over all functions f_k (with fixed k) and hypotheses h, Equation 28 gives the distribution of error histograms. In this equation, we simply factorize ν; $P(\epsilon\,|\nu = (\nu_1, \ldots, \nu_{2^i}))$ is quantified by Equation 26, and $P(\nu = (\nu_1, \ldots, \nu_{2^i}))$ by Equation 25.

$$\pi_k(\epsilon|H_i) = \sum_\nu P(\epsilon\,|\nu = (\nu_1, \ldots, \nu_{2^i}))P(\nu = (\nu_1, \ldots, \nu_{2^i})) \quad (28)$$

Finally, we can quantify the expected (over all samples S and target functions f_k) resulting error rate in Equation 29.

$$E_{\{f_k,S\}}(\epsilon(h_L)|H_i, m, k) = \sum_\nu \left(\frac{\int \epsilon \left(\sum_e P(h_\epsilon \in H_i^*|e(h_\epsilon) = e, m)B[\epsilon, m](e)P(\epsilon|\nu)\right) d\epsilon}{\int \sum_e P(h_\epsilon \in H_i^*|e(h_\epsilon) = e, m)B[\epsilon, m](e)P(\epsilon|\nu) d\epsilon} P(\nu = (\nu_1, \ldots, \nu_{2^i})) \right) \quad (29)$$

$P(h_\epsilon \in H_i^*|e(h_\epsilon) = e, m)$ is quantified by Equation 24, $P(\epsilon|\nu)$ by Equation 26, and $P(\nu = (\nu_1, \ldots, \nu_{2^i}))$ by Equation 25. We can evaluate Equation 29 easily as it refers only to the binomial distribution, the sample size and the numbers of attributes i and k.

(2): $i \geq k$. In this case, the target function assigns one class label to 2^{i-k} instances which can be distinguished by the hypothesis. The hypothesis distinguishes 2^i subspaces; a randomly drawn hypothesis will assign each of these subspaces the correct class label half the time. Hence, the distribution of error rates is governed by the binomial distribution as given in Equation 30.

$$\pi(\epsilon|H_i, f_k) = B\left[\frac{1}{2}, 2^i\right] \quad (30)$$

We can quantify the expected resulting error in Equation 31 by replacing π in Equation 20 by the binomial distribution.

$$E_{\{f_k,S\}}(\epsilon(h_L)|H_i, m, k) \quad (31)$$

$$= \frac{\int \epsilon \left(\sum_e P(h_\epsilon \in H_i^*|e(h_\epsilon) = e, m) P(e(h_\epsilon) = e|\epsilon, m) B[\frac{1}{2}, 2^i](\epsilon)\right) d\epsilon}{\int_\epsilon \sum_e P(h_\epsilon \in H_i^*|e(h_\epsilon) = e, m) P(e(h_\epsilon) = e|\epsilon, m) B[\frac{1}{2}, 2^i](\epsilon) d\epsilon}$$

$P(h_\epsilon \in H_i^*|e(h_\epsilon) = e, m)$ is given by Equation 24. Let us check whether Equations 29 and 31 predict the error rate of a learner accurately. In our experiments, we drew 200 Boolean functions with 3 relevant attributes and allowed model classes of between 1 and 6 attributes. Figure 1 shows the averaged error histograms for all model classes. Figure 2 compares theoretical and measured error rates $\epsilon(h_i^L)$ of hypotheses with least empirical error. We can see that the predicted error rates fit the measured rates fairly closely.

Note that the averaged error histograms of model classes 1 through 3 are equal. As long as the error histogram stays constant, increasing the number of hypotheses decreases the resulting error rate. As we add irrelevant attributes, the ratio of hypotheses with very low error rates decreases and the resulting error increases.

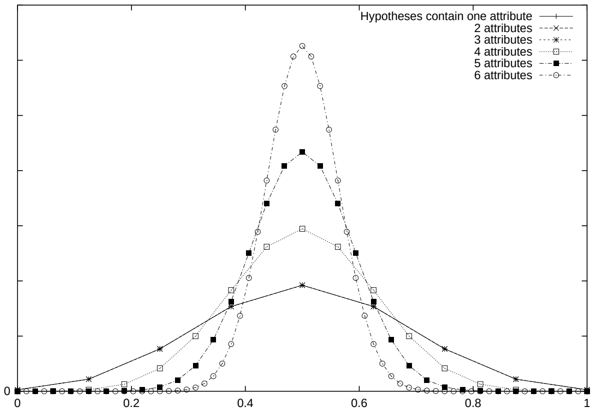

Fig. 1. Error histograms for models which contain Boolean attributes $x_1, \ldots x_i$ when the target function requires attributes x_1, x_2, x_3. The distributions are equal in the first three models; the variance then increases.

5 Decision Trees and Unknown Targets

In general, the error histogram is not known. However, we can estimate the error histogram from the sample and thus apply the analysis approximately for arbitrary target distributions. As an estimate of $\pi(\epsilon|H_i)$ we use the empirical counterpart $\pi(e|H_i)$ (the distribution of empirical error rates of hypotheses in H_i with respect to the sample S) which we can record when H_i is known and a sample S is available. We can obtain $\pi(e|H_i)$ by repeatedly drawing hypotheses

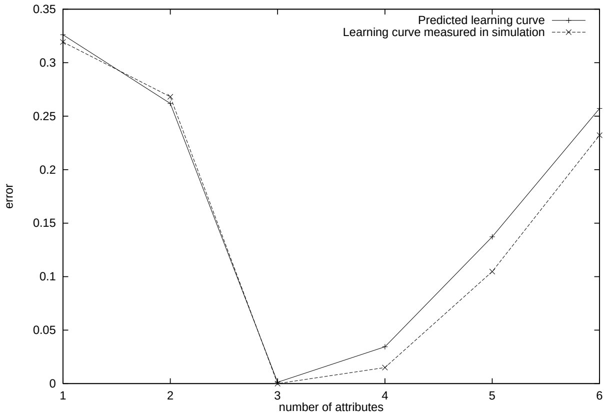

Fig. 2. (b) Learning curve: Expected error (theoretical and measured values) when the target function requires attributes x_1 through x_3 and model H_i (i is on the horizontal axis) uses attributes x_1 through x_i.

from H_i under uniform distribution, or by conducting a Markov random walk in the hypothesis space with the uniform distribution as stationary distribution [4].

This raises the question whether estimating the error histogram of a model class sufficiently accurately is any easier than estimating the error rate of all hypotheses in that model class. Fortunately, Langford and McAllester [8] have answered this question affirmatively. It is obvious that the empirical error histogram converges toward the true error histogram when m grows – in other words, $\lim_{m\rightarrow\infty} P(e|H_i) = \pi(\epsilon|H_i)$. However, when m goes to infinity, then all empirical error rates converge to their corresponding true error rates and the error prediction problem becomes trivial as we can treat the training sample error rates as true error rates. One of the main results of PAC theory (*e.g.,* [6]) is that we achieve uniform convergence (*i.e., all* empirical error rates approximate their corresponding true error rates accurately) only when $\frac{\log|H_i|}{m}$ is sufficiently small. However, the empirical error histogram converges to the true histogram even if $\frac{\log|H_i|}{m}$ is arbitrarily large.

Consider a process in which both the sample size m_i and the size of the model class grow in parallel when $i \rightarrow \infty$, such that $\frac{\log|H_i|}{m_i}$ stays constantly large. Over this process, we are unable to estimate all error rates in H_i but $P(e|H_i)$ converges to $\pi(\epsilon|H_i)$ as i grows [8]. In this respect, estimating the histogram is much easier than estimating all error rates in H_i. For an extended discussion on the complexity and accuracy of estimating π, see [14].

The objective of the next experiment is to check whether our analysis can predict the error rate of a decision tree learner accurately for a set of problems from the UCI data set repository. For each problem and every number of leaf nodes i, we estimate the histogram of error rates $\pi(\epsilon|H_i)$ using 4000×2^i randomly drawn decision trees using an algorithm described in [14] running in $O(4000i)$. Using the estimate of π, we evaluate Equation 20. We also run a decision tree learner that minimizes the empirical error rate using exactly i leaf nodes [15]. We use the resulting empirical error to evaluate Equation 7. We then run a 10-fold cross validation loop (for each number i). In each fold, we run the exhaustive/greedy learner and estimate the generalization error using the holdout set.

Figure 3 compares the predicted to the measured generalization error rates (based on Equation 20) for the empirical error minimizing learner learner, and Figure 4 compares predicted error given the empirical error (Equation 7) to measured error. For most measurements, the predicted value lies within the standard deviation of the measured value which indicates that the predictions are relatively accurate. Only for the Cleveland and *E. Coli* problem we can see significant deviations.

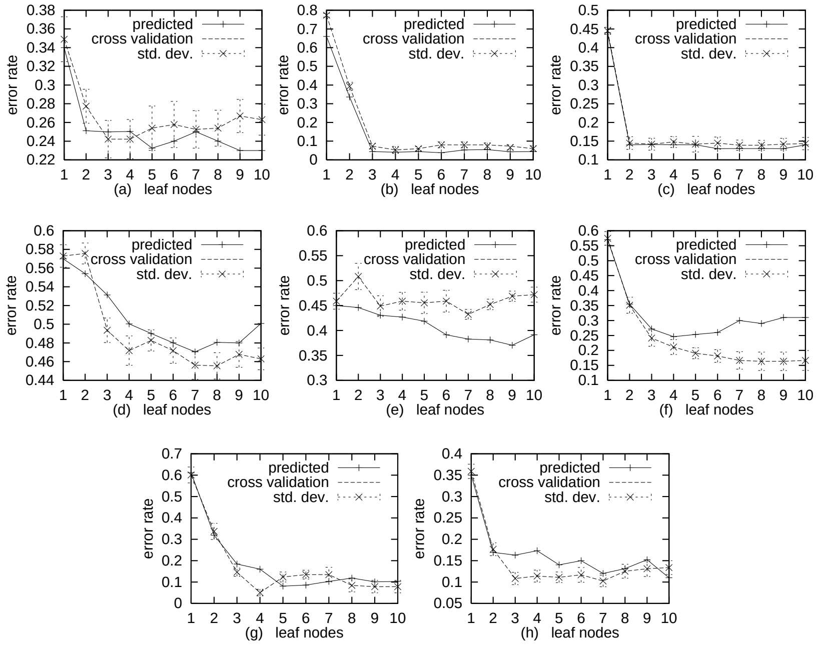

Fig. 3. Predicted (Equation 20) and measured (10-fold cross validation) generalization error rates of decision trees restricted to i leaf nodes. (a) diabetes, (b) iris, (c) crx, (d) cmc, (e) cleveland, (f) ecoli, (g) wine, (h) ionosphere.

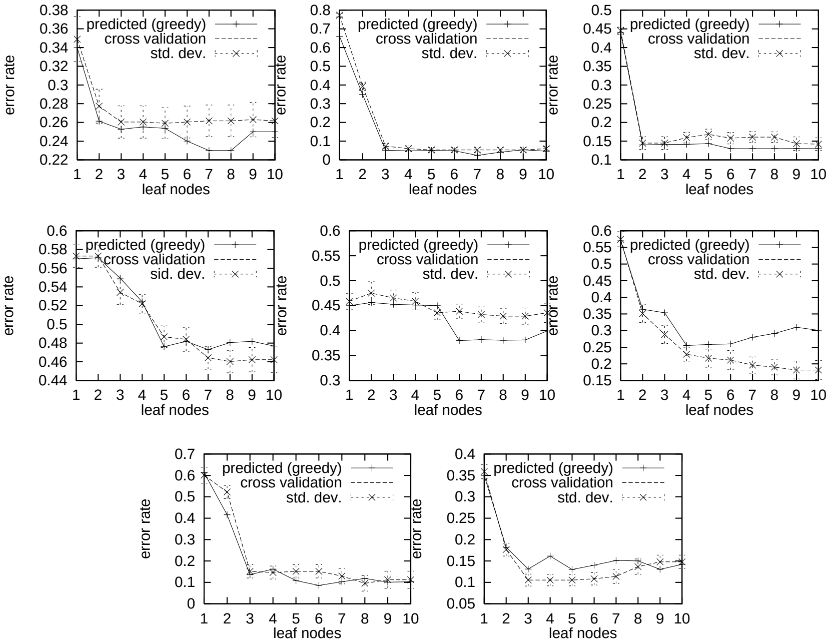

Fig. 4. Predicted (Equation 7) and measured (10-fold cross validation) generalization error rates of a decision tree learner (based on measured empirical error rates), restricted to i leaf nodes. (a) diabetes, (b) iris, (c) crx, (d) cmc, (e) cleveland, (f) ecoli, (g) wine, (h) ionosphere.

6 Discussion

Average-case analyses quantify the expected (over all samples) error of a learning algorithm for *a* given target function. Consequently, they are able to predict the behavior of a learning algorithm for a specific learning problem much better than worst-case analyses. Unfortunately, average-case analyses are not quite as easy to apply as worst-case analyses. The reason is their reference to specific properties of the underlying learning problems which typically are not known. In science, this corresponds to the initial state of a physical system that has to be known before the development of that system over time can be predicted.

In most cases, average-case analyses break the error rate only approximately into measurables and domain properties. This is clearly a drawback, but it does not automatically void the usefulness of such analyses. Since the strength of such approximations is often difficult to quantify, in most cases the only feasible way is to run learning algorithms and to measure the deviation between predicted and measured error rates. The experiments presented in this paper provide evidence for the usefulness of the approximate Equation 20. The analysis of the error rate given the empirical error (Equation 7) differs from most known analyses by not being approximate.

Average-case analyses have been discussed for various learners. Iba and Langley [7] have studied the behavior of decision stump learners. Okamoto and Yugami [11, 12] presented an analysis for k-nearest neighbor classifiers; Fukumizu [3] for linear neural networks. Reischuk and Zeugmann [13] analyzed the average time complexity of an algorithm that learns one-variable pattern languages. An analysis of Naive Bayesian classifiers has been presented by Langley *et al.* [10]; under some simplifying approximations [9] the analysis becomes computationally efficient. An average-case analysis of cross validation has been presented in [16].

A first version of the analysis class was presented by Scheffer and Joachims [18, 17] and later generalized [19] and applied to text categorization and decision tree regularization [15]. Independently, Domingos [1] presented a similar analysis which additionally assumes that all hypotheses incur equal error rates. Lifting the latter assumption [2] leads to an analysis that (besides making the additional assumption that the training set error is known) deviates from the first analysis [18] only in some technical details.

The histogram of error rates has been used to improve on *worst-case* error bounds. The idea of a worst-case analysis of [5] is that hypotheses with an error rate of much more than the desired error bound ε have a much smaller chance of incurring the least empirical error than hypotheses with an error rate that lies just slightly above ε. In contrast to the resulting *shell decomposition bounds*, we obtain the exact distribution that governs the resulting error rate (and therefore also the expected error).

An interesting question to pose is whether the estimated empirical error histogram can lead to a non-approximate claim on the resulting generalization error. Given the uncertainty that remains when the histogram has been estimated, it is not possible to determine the *exact expected* generalization error (which we are concerned about in this paper), but Langford and McAllester [8] have proven worst-case shell decomposition *bounds* that differ from those of [5] by taking into account that the histogram is only estimated.

We have shown that the error histogram for Boolean functions is a certain binomial distribution. A fundamental question is whether there is a more general link between the error histogram and measurable properties (such as the VC dimension) of the model class and the class of target functions.

References

1. P. Domingos. A process-oriented heuristic for model selection. In *Proceedings of the Fifteenth International Conference on Machine learning*, pages 127–135, 1998.
2. P. Domingos. Process-oriented estimation of generalization error. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligenct*, 1999.
3. K. Fukumizu. Generalization error of linear neural networks in unidentifiable cases. In *Proceedings of the Tenth International Conference on Algorithmic Learning Theory*, 1999.

4. W. Gilks, S. Richardson, and D. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice.* Chapman & Hall, 1995.
5. D. Haussler, M. Kearns, S. Seung, and N. Tishby. Rigorous learning curve bounds from statistical mechanics. *Machine Learning*, 25, 1996.
6. David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, September 1992.
7. W. Iba and P. Langley. Induction of one-level decision trees. In *Proceedings of the Ninth International Conference on Machine Learning*, pages 233–240, 1992.
8. J. Langford and D. McAllester. Computable shell decomposition bounds. In *Proceedings of the International Conference on Computational Learning Theory*, 2000.
9. P. Langley and S. Sage. Tractable average case analysis of naive bayes classifiers. In *Proceedings of the Sixteenth International Conference on Machine Learning*, pages 220–228, 1999.
10. Pat Langley, Wayne Iba, and Kevin Thompson. An analysis of bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 223–228, 1992.
11. S. Okamoto and Y. Nobuhiro. An average-case analysis of the k-nearest neighbor classifier for noisy domains. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence*, pages 238–243, 1997.
12. S. Okamoto and N. Yugami. Generalized average-case analysis of the nearest neighbor algorithm. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pages 695–702, 2000.
13. Rüdiger Reischuk and Thomas Zeugmann. Learning 1-variable pattern languages in linear average time. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory*, pages 198–208, 1998.
14. T. Scheffer. *Error Estimation and Model Selection.* Infix Publisher, Sankt Augustin, 1999.
15. T. Scheffer. Nonparametric regularization of decision trees. In *Proceedings of the European Conference on Machine Learning*, 2000.
16. T. Scheffer. Predicting the generalization performance of cross validatory model selection criteria. In *Proceedings of the International Conference on Machine Learning*, 2000.
17. T. Scheffer and T. Joachims. Estimating the expected error of empirical minimizers for model selection. Technical Report TR 98-9, Technische Universitaet Berlin, 1998.
18. T. Scheffer and T. Joachims. Estimating the expected error of empirical minimizers for model selection (abstract). In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 1998.
19. T. Scheffer and T. Joachims. Expected error analysis for model selection. In *Proceedings of the Sixteenth International Conference on Machine Learning*, 1999.