Nonparametric Regularization of Decision Trees*

Tobias Scheffer

Otto von Guericke University, FIN/IWS, Universitätsplatz 2,
39106 Magdeburg, Germany
scheffer@iws.cs.uni-magdeburg.de

Abstract. We discuss the problem of choosing the complexity of a decision tree (measured in the number of leaf nodes) that gives us highest generalization performance. We first discuss an analysis of the generalization error of decision trees that gives us a new perspective on the regularization parameter that is inherent to any regularization (*e.g.,* pruning) algorithm. There is an optimal setting of this parameter for every learning problem; a setting that does well for one problem will inevitably do poorly for others. We will see that the optimal setting can in fact be estimated from the sample, *without* "trying out" various settings on hold-out data. This leads us to a nonparametric decision tree regularization algorithm that can, in principle, work well for all learning problems.

1 Introduction

Decision tree algorithms (*e.g.,* [14, 3]) have to solve two distinct problems: they need to identify the size of the tree that leads to optimal generalization performance and, subject to these size constraints, they have to minimize the empirical error rate. The problem of choosing the appropriate tree size is in essence a problem of estimating the misclassification probability of the best decision tree of a given size.

A quick clarification of some notational details is useful for further discussion. Let H_i be the class of all decision trees with exactly i leaf nodes over some fixed set of possible tests. $h \in H_i$ is then a decision tree and maps instances x to class labels y. A learning 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 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.

Many decision tree algorithms try to minimize the generalization error by minimizing a regularization function $f(e(h), c(H_i))$ that depends on the empirical error $e(h)$ and some complexity measure $c(H_i)$ of the hypothesis class H_i

* European Conference on Machine Learning (ECML-2000). Barcelona, Spain, 2000.
© Springer-Verlag

which the hypothesis tree h came from. In other words, the complexity (or size) of the decision tree is getting penalized. Technically, this is often realized by employing some pruning rule that trades off a lower empirical error rate against the number of branches required to achieve this gain in empirical accuracy. Such complexity regularization techniques will in fact lead to a low generalization error rate *if and only if* $f(e(h), c(H_i))$ is a "reasonable" estimate of $\epsilon(h)$, the generalization error rate. This raises the question whether *there is* a regularization function $f(e, c(H_i))$ that maps an empirical error rate and some measure of the complexity of a class of decision trees H_i to a "reasonable" estimate of $\epsilon(h)$.

We can use (PAC-style) Chernoff bounds to bound the greatest possible difference between true and empirical error rate of any hypothesis in H_i and then conclude that, with high probability, the generalization error rate of h is no more than $e(h) + \mathit{bound}(m, |H_i|)$, where $|H_i|$ is the size (alternatively, the VC-dimension) of H_i. However, the actual error rate may lie *anywhere* between 0 and the worst-case bound, depending on characteristics of the given learning problem. We can easily construct two learning problems (one with an error rate that increases steeply when $|H_i|$ grows and one with a slowly increasing error curve) such that any regularization function $f(e, c(H_i))$ fails (*i.e.,* incurs an additional error of $\lambda > 0$ that does not vanish when the sample size grows) for at last one of them [9]. This means that the empirical error rate $e(h)$ and the complexity of H_i do not suffice to determine the actual error rate; some information is missing. Obviously, we can determine a near optimal setting for the regularization parameter for each single problem by trying out many values and assessing the resulting decision tree on holdout data. Alternatively, we can use cross validation to select the optimal number of leaves in the first place, like, for instance, the CART algorithm does [3]. The primary disadvantage of n-fold cross validation [19] lies in its unsatisfactory efficiency caused by the necessity of invoking the learner n times for each considered number of leaves.

We will pursue a different approach. We will take a careful look at the generalization error rate of decision trees and study just what information regularization functions are missing – *i.e.,* what other information than $e(h)$ and the complexity of H_i do we need o obtain a reliable estimate of $\epsilon(h)$ for all possible problems, without assessing hypotheses on holdout data. We will identify this missing information and discuss how it can be acquired efficiently in many cases. In Section 2, we simplify the expected error analysis of [18] slightly and apply it to the problem of choosing the optimal decision tree complexity. The original analysis is restricted to exhaustive learners while decision tree algorithms are usually greedy. Our main theoretical result (Section 3) is an extension of the analysis to greedy learning algorithms. In Section 4 we discuss a nonparametric regularization algorithm which we study empirically in Section 5.

2 Error Rate of Exhaustive Decision Tree Learners

In this section, we assume that, given a number i of leaf nodes, the learning algorithm determines the hypothesis h_i^L with least *empirical* error that has exactly

i leaf nodes. When there are several hypotheses with the same low empirical error, we assume the learner to break ties by drawing at random under uniform distribution. Let us assume that the sample size m is fixed and given in advance, whereas the sample S itself is a random variable, governed by the distribution $(p(x,y))^m$. We will now study $E(\epsilon(h_i^L)|H_i,m)$, the expected *generalization* error of the returned hypothesis h_i^L given the sample size m and the number of leaves i. In order to determine the expected true error (expected over all samples) of h_i^L (the decision tree with i leaf nodes that incurs the least empirical error rate), we factorize the hypothesis h that the learner returns (Equation 1). 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 1 such that all hypotheses h_ϵ with true error ϵ are grouped together. $\pi(\epsilon|H_i)$ is the density of decision trees with error rate ϵ among all the decision trees with i leaf nodes (with respect to the given learning problem). Intuitively, if we would draw a decision tree with i leaf nodes at random under uniform distribution from all decision trees H_i, $\pi(\epsilon|H_i)$ would be the chance of the resulting decision tree incurring an error rate of ϵ for the given problem. This takes us to Equation 2.

$$E(\epsilon(h_i^L)|H_i,m) = \int_h \epsilon(h)P(h_i^L = h|H_i,m)dh \tag{1}$$
$$= \int_\epsilon \epsilon P(h_i^L = h_\epsilon|\epsilon, H_i, m)\pi(\epsilon|H_i)d\epsilon \tag{2}$$

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 with respect to some sample S. 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 3). A hypothesis that does not lie in H_i^* has a zero probability of becoming h_i^L (Equation 4). In Equation 5, 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 6). In Equation 7, we factorize the least empirical error e and, in Equation 8, we simply split up the conjuction (like $p(a,b) = p(a)p(b|a)$).

$$P(h_L = h_\epsilon|\epsilon, H_i, m)$$
$$= P(h_L = h_\epsilon|H_i, m, h_\epsilon \in H_i^*)P(h_\epsilon \in H_i^*) \tag{3}$$
$$+P(h_L = h_\epsilon|H_i, m, h_\epsilon \notin H_i^*)(1 - P(h_\epsilon \in H_i^*))$$
$$= P(h_L = h_\epsilon|H_i, m, h_\epsilon \in H_i^*)P(h_\epsilon \in H_i^*) \tag{4}$$
$$= \sum_n P\big(h_L = h_\epsilon \big| H_i, m, h_\epsilon \in H_i^*, |H_i^*| = n\big) P(h_\epsilon \in H_i^*, |H_i^*| = n) \tag{5}$$
$$= \sum_n \frac{1}{n} P(h_\epsilon \in H_i^*, |H_i^*| = n) \tag{6}$$

$$= \sum_e \sum_n \frac{1}{n} P\big(h_\epsilon \in H_i^*, |H_i^*| = n \big| e(h_\epsilon) = e\big) P(e(h_\epsilon) = e|\epsilon, m) \tag{7}$$

$$= \sum_e \sum_n \frac{1}{n} P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) P\big(|H_i^*| = n \big| h_\epsilon \in H_i^*, e(h_\epsilon) = e\big)$$
$$P(e(h_\epsilon) = e|\epsilon, m) \tag{8}$$

By inserting Equation 8 into Equation 2 we get

$$E(\epsilon(h_L)|H_i, m) \tag{9}$$
$$= \int_\epsilon \epsilon \Bigg(\sum_e \sum_n \frac{1}{n} P\big(|H_i^*| = n \big| h_\epsilon \in H_i^*, e(h_\epsilon) = e\big) P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m)$$
$$P(e(h_\epsilon) = e|\epsilon, m) \pi(\epsilon|H_i) \Bigg) d\epsilon \tag{10}$$

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 $const = P\big(|H_i^*| = n \big| h_\epsilon \in H_i^*, e(h_\epsilon) = e\big)$ is constant for all h_ϵ. Equation 10 specifies the expectation of $\epsilon(h_L)$. The density $p(\epsilon(h_L))$ has to integrate to 1. *const* is therefore a normalization constant which is determined uniquely.

$$const = \left(\int_\epsilon \sum_e P(h_\epsilon \in H_2^* | e(h_\epsilon) = e, m) P(e(h_\epsilon) = e|\epsilon, m) \pi(\epsilon|H_i) d\epsilon \right)^{-1} \tag{11}$$

Combining Equations 10 and 11 we obtain

$$E(\epsilon(h_L)|H_i, m)$$
$$= \frac{\int_\epsilon \epsilon \left(\sum_e P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) P(e(h_\epsilon) = e|\epsilon, m) \pi(\epsilon|H_i) \right) d\epsilon}{\int_\epsilon \sum_e P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) P(e(h_\epsilon) = e|\epsilon, m) \pi(\epsilon|H_i) d\epsilon} \tag{12}$$

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 2). Each of these hypotheses incurs an empirical error rate that is by itself governed by the binomial distribution $B[m, \epsilon]$. (Each example can be classified correctly or erroneously; the chance of the latter happening is ϵ; this leads to a binomial distribution). Let us assume 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 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

$$P(h_\epsilon \in H_i^* | e(h_\epsilon) = e, m) = \prod_{\epsilon'} P(e(h) \geq e | \epsilon', m)^{|H_i| \pi(\epsilon' | H_i)}. \tag{13}$$

Finally, let us determine $|H_i|$, the number of decision trees with i leaf nodes. With c classes and a total of n possible tests available, there are exactly $|H_i| = \tau(i) \times c^i$ decision trees with i leaf nodes ($\tau(i)$ is the number of "trunks" and there are c^i labelings of the leaf nodes), where $\tau(1) = 1$ and $\tau(i) = \sum_{j=1}^{i-1} n \times \tau(j) \times \tau(i-j)$. Intuitively, at each test node there are n possible tests and j of the remaining i leaf nodes can be placed in the left subtree while the remaining $j - i$ leaf nodes go into the right subtree (for all possible j between 1 and $i - 1$).

What have we achieved so far? Equations 12 and 13 quantify the expected generalization error of h_i^L for a given problem in terms of three quantities: the number of decision trees $|H_i|$ (can easily be computed), the sample size m (which is known), and the density of error rates in H_i, $\pi(\epsilon|H_i)$. Note that, for Equations 12 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 12 and 13 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 which motivates the negative result of Kearns *et al.* [9] on the performance of complexity regularization algorithms. 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. We have therefore identified the information that complexity penalization algorithms are missing as being $\pi(\epsilon|H_i)$. If there was a feasible way to estimate $\pi(\epsilon|H_i)$ we could construct a regularization algorithm that uses this additional information and circumvents the negative result of Kearns *et al.*. But before we discuss how π can be estimated, let us look at the generalization error of greedy decision tree learners.

3 Greedy Decision Tree Algorithms

The solution presented so far quantifies the generalization error of the decision tree with i leaf nodes that incurs the least empirical error with respect to a sample of size m. Hence, the analysis applies to exhaustive learners that are able to always find the empirical error minimizing hypothesis. However, when the problem requires the decision tree to have many nodes, exhausting the space of all decision trees with that number of nodes may not be feasible and a greedy algorithm (that cannot be guaranteed to find the decision tree with least empirical error) may have to be employed. We will now discuss the expected generalization error of a hypothesis with an empirical error rate of e, (found, for instance, by a greedy learner) which may be distinct from the hypothesis with the globally smallest empirical error rate. The following solution depends additionally on the empirical error rate e of the hypothesis returned by the learner. This means that we have to run the greedy learner and determine the resulting training set error which was not necessary in the exhaustive analysis.

Let H_i^e be the subset of H_i with empirical error of $e(h) = e$. Let h_i^e be a hypothesis drawn from H_i^e at random under uniform distribution – *i.e.,* h_i^e is an arbitrary hypothesis with empirical error rate of e. We start off by factorizing the hypothesis which the learner chooses as h_i^e (Equation 14). Similarly to Equation 2, in Equation 15 we factorize the error rate ϵ, forming (for each ϵ) "subgroups" of $\pi(\epsilon|H_i)$ hypotheses with equal error rate ϵ. In Equation 16 we factorize the empirical error rate of h_i^e and then say that all empirical error rates have probability zero, except for the value e which has a probability of 1. In Equation 17, we factorize the cardinality of H_i^e and, in Equation 18, we claim that the chance of a hypothesis h_ϵ in H_i^e being selected as h_i^e is $\frac{1}{n}$ when $|H_i^e| = n$ (remember that we assumed h_i^e to be drawn at random from H_i^e).

The probability of H_i^e being of size n when we know already that one hypothesis (h_i^e) is in this set is equal to the chance of H_i^e being of size $n-1$ (Equation 19), and the empirical error rate of h_ϵ is governed by the binomial distribution with mean value ϵ. Now note that the sum over n in Equation 20 does not depend on ϵ any more – *i.e.,* it is a constant. Since $p(\epsilon(h_i^e)|H_i, m, e)$ has to integrate to 1 can simply normalize the expectation in Equation 21 like we did in Equation 11. Now only $\pi(\epsilon|H_i)$ remains which means that we are done.

$$
\begin{aligned}
&E(\epsilon(h_i^e)|H_i, m, e) \\
&= \int \epsilon(h) P(h_i^e = h|H_i, m, e) dh && (14) \\
&= \int \epsilon P(h_i^e = h_\epsilon|\epsilon, H_i, m, e)\pi(\epsilon|H_i) d\epsilon && (15) \\
&= \int \epsilon P(h_i^e = h_\epsilon|\epsilon, H_i, m, e) P(e(h_\epsilon) = e)\pi(\epsilon|H_i) d\epsilon && (16) \\
&= \int \epsilon \sum_n P\left(h_i^e = h_\epsilon \Big| |H_i^e| = n, \epsilon, H_i, m, e\right) \\
&\qquad\qquad P\left(e(h_\epsilon) = e, |H_i^e| = n\right)\pi(\epsilon|H_i) d\epsilon && (17) \\
&= \int \epsilon \sum_n \frac{1}{n} P(e(h_\epsilon) = e) P\left(|H_i^e| = n \Big| e(h_\epsilon) = e\right)\pi(\epsilon|H_i) d\epsilon && (18) \\
&= \int \epsilon \sum_n \frac{1}{n} P(e(h_\epsilon) = e) P\left(|H_i^e| = n-1\right)\pi(\epsilon|H_i) d\epsilon && (19) \\
&= \int \epsilon B[\epsilon, m](e) \left(\sum_n \frac{1}{n} P\left(|H_i^e| = n-1\right)\right)\pi(\epsilon|H_i) d\epsilon && (20) \\
&= \frac{\int \epsilon B[\epsilon, m](e)\pi(\epsilon|H_i) d\epsilon}{\int B[\epsilon, m](e)\pi(\epsilon|H_i) d\epsilon} && (21)
\end{aligned}
$$

We have now found a solution that quantifies $E(\epsilon(h_i^e)|H_i, m, e)$, the expected generalization error of a decision tree with i leaf nodes and empirical error rate e for a given learning problem $p(x, y)$. In contrast to the result of Section 2, h_i^e is not assumed to be the result of an exhaustive learner, it can be the outcome of a greedy learner. This time, the solution depends on the empirical error e (*i.e.,* we need to run the learner to determine the training set error), the density of error rates in the set of decision trees with i leaf nodes, $\pi(\epsilon|H_i)$, and the sample size m, but it does not depend on $|H_i|$. Again, Equation 21 specifies the actual error rate for the given learning problem rather than a worst-case bound that holds for all possible learning problems (which PAC theory does). The additional information of $\pi(\epsilon|H_i)$ makes this possible.

4 Nonparametric Decision Tree Regularization

Decision tree pruning algorithms minimize a regularization function $f(e(h_i^L), c(H_i))$ (depending on the empirical error rate and some complexity measure of H_i) which has to be a good estimate of $\epsilon(h_i^L)$ if we want to minimize the right quantity. However, the influence of the complexity on the error has to be weighted and this weight has to be chosen for each problem. If, however, $\pi(\epsilon|H_i)$ was known, then we could construct a decision tree learner that minimizes Equation 12 (for exhaustive learning) or Equation 21 (for greedy learning), respectively. We then have a regularization function that, in principle, should work well for all possible learning problems without having a parameter.

$\pi(\epsilon|H_i)$ cannot be measured directly since it depends on $p(x, y)$ which is unknown. However, there is an empirical counterpart $\pi(e|H_i)$ (the density 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 from H_i under uniform distribution, or by conducting a Markov random walk in the hypothesis space with the uniform distribution as stationary distribution [7, 12]. While the general problem of estimating densities is very hard, the situation is not quite as bad in our special case. Like $\pi(\epsilon|H_i)$, $\pi(e|H_i)$ is one-dimensional, but is is furthermore discrete since there are only $m + 1$ possible empirical error rates when m is the sample size. How many hypotheses of H_i do we have to look at in order to obtain a reliable estimate of π? We want to estimate m probabilities; suppose that we want none of these estimates to be off by more than some ε with high probability $(1 - \delta)$. In this case, we need to draw $\frac{1}{2\varepsilon^2} \log \frac{m}{\delta} = O(\log m)$ hypotheses which is not particularly much. Although drawing $O(\log m)$ hypotheses will typically suffice for an accurate estimate of $\epsilon(h_i^L)$, there are cases in which a misestimation of π by some small ε can lead to an inaccurate estimate of $\epsilon(h_i^L)$. In this theoretical worst-case, estimating π sufficiently accurately can be as difficult as running a learning algorithm, see [15] for a more detailed discussion. We can now describe a decision tree algorithm that uses the expected error analysis to regularize the decision tree complexity.

Algorithms QDT and Greedy-QDT.

1. **For** $i = 1 \ldots \texttt{maxleaves}$.
   (a) Draw $O(\log m)$ decision trees with i leaf nodes and record their empirical error rates, thus measuring $\pi(e|H_i)$ which will serve as an estimate of $\pi(\epsilon|H_i)$.
   (b) For exhaustive **QDT**: Evaluate Equation 12 to determine the estimated expected error of $\epsilon(h_i^L)$.
   (c) For **Greedy-QDT**: Minimize the empirical error greedily using exactly i leaf nodes, the resulting hypothesis is h_i^L. Determine the empirical error $e(h_i^L)$ on the training set. Evaluate Equation 21 to obtain an estimate of the expected generalization error of h_i^L.
2. Let $i*$ be the number i which minimizes the estimated expected generalization error of h_i^L determined in step 1b or 1c, respectively.
3. For exhaustive **QDT**: Exhaust the space of decision trees with $i*$ leaf nodes (this takes $O(n^{i*})$). The resulting tree is h_{i*}^L.
4. For **Greedy-QDT**: h_{i*}^L has already been determined.
5. **Return** h_{i*}^L.

For a given number of leaf nodes, we use the following algorithm to minimize the empirical error rate. When we set the threshold to i, the algorithm is almost exhaustive while with a threshold of 1 it is completely greedy. We use the information gain heuristic; note, however, that our complexity regularization method can be "plugged" into almost any greedy or exhaustive decision tree learner.

Algorithm EmpiricalErrorMinimization

1. **Input:** Number i of nodes, **Output:** decision tree with least empirical error.
2. **If** $i = 1$ return leaf node with class label that minimizes the empirical error rate.
3. **For** all attributes a,
   (a) Find optimal split for the given attribute a,
   (b) **If** $i >$ *threshold* commit to the split. **Otherwise** backtrack to find the globally optimal split.
   (c) **If** $i >$ *threshold* **Then Let** $left : right = H_{left} \times$ # of instances in the left branch : $H_{right} \times$ # of instances in right ranch (split the number of remaining nodes according to the remaining entropy weighted by the number of instances in the left and right branch).
   (d) **Otherwise** backtrack to find optimal values for $left$ and $right$.
      i. Determine left and right subtree by invoking **EmpiricalErrorMinimization** recursively with $left$ and $right$ as desired number of leaf nodes, and with the corresponding subset of the sample.

Some technical details are left for the full paper, due to lack of space. An algorithm that records the error rates of $n \times |Y|^i$ decision trees in $O(n \times i \times m)$ (required for step 1a) is described in [15].

5 Empirical Studies

In order to select an appropriate number of leaf nodes, the QDT algorithm has to be able to predict the error rate in dependence of the number of leaf nodes used. Therefore, in the first part of the empirical studies, we will study how the predicted error rate (depending on the number of leaves) relate to the error rates measured by n-fold cross validation.

Is the error rate predicted accurately? We drew 8 learning problems that have little (or no) missing values at random from the UCI repository [2]. For each problem and every number of leaf nodes i, we estimate the density of error rates $\pi(\epsilon|H_i)$ in $O(i \times m)$. For the exhaustive version (results in Figure 1), we evaluate Equation 12 to obtain the predicted generalization error. For the greedy version (results in Figure 2), we run the greedy learner, measure the empirical error e and evaluate Equation 21 to get the predicted generalization error. 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 (the exhaustive learner is not completely exhaustive; due to the high computational costs only subtrees of up to four leaves are searched exhaustively). Figure 1 compares the predicted to the measured generalization error rates for the exhaustive learner and Figure 2 for the greedy learner. For most measurements, the predicted value lies within the standard deviation of the measured value which indicates that the predictions are accurate. Note that, even if all predictions were totally accurate, 14% of all predictions would lie outside their standard deviation. Only for the Cleveland and *E. Coli* problem we can see significant deviations; but there is no case in which relying on the prediction would result in selecting a number of leaves that is significantly suboptimal. In some cases, the greedy analysis appears to give just slightly more accurate predictions. This might be due to the fact that the greedy analysis gets to know the resulting empirical error rate as additional information. In many cases, the exhaustive learner achieves a slightly (not significantly) lower generalization error.

Does QDT better or worse than cross validation based learning? We will now study how our regularization procedure compares to cross validation based pruning (*e.g.,* [3]). We wrap the learner into an outer layer of 10-fold cross validation. In this "wrapper", for every number of leaf nodes i we first evaluate Equation 12 to estimate which number of leaves would be optimal. We then run an inner loop of n-fold cross validation to find out which number of leaf nodes the cross validation based learner would select. We assess both recommended numbers of leaf nodes in the outer cross validation wrapper. We try this for various n. Figure 3 shows the results. Surprisingly, using 10-fold cross validation is in no case significantly better than using one fold (training and test with a split ratio of 70%). In some cases (*e.g.,* wine) this might still be the case but the differences are not significant. The differences between QDT and 10-fold cross validation based selection of the number of leaf nodes are not significant – in other words, our analysis determines the optimal number of leaf nodes just as good as n-fold cross validation (for the studied problems).

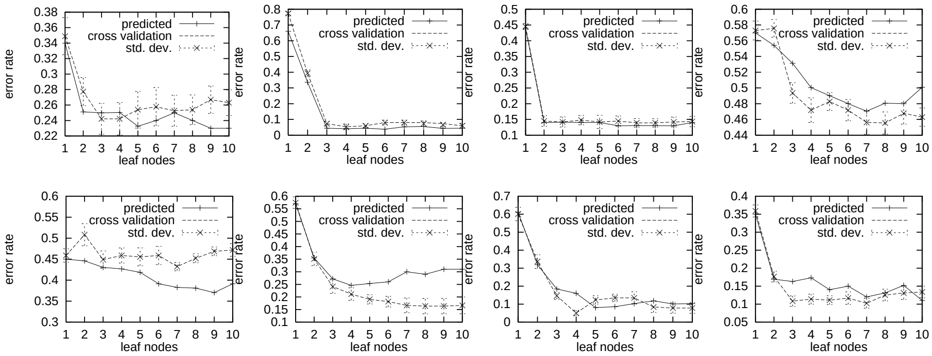

Fig. 1. Predicted (expected error analysis) and measured (10-fold cross validation) generalization error rates of decision trees restricted to i leaf nodes. (a) diabetes, (b) iris, (c) crx, (d) cmc, (e) cleveland, (f) ecoli, (g) wine, (h) ionosphere.

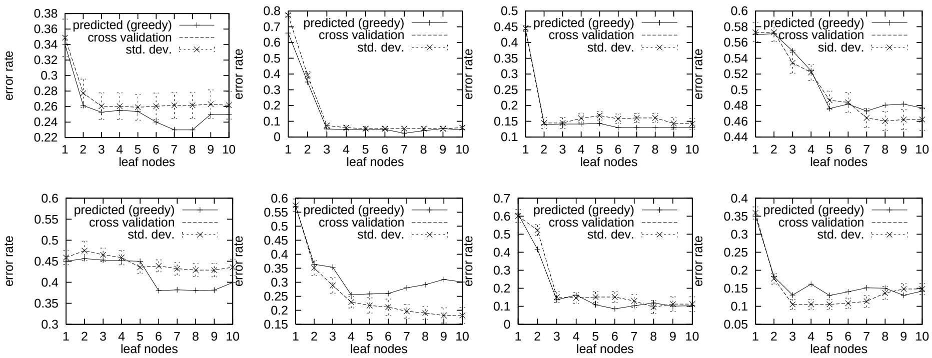

Fig. 2. Predicted and measured (10-fold cross validation) generalization error rates of a greedy decision tree learner, restricted to i leaf nodes. (a) diabetes, (b) iris, (c) crx, (d) cmc, (e) cleveland, (f) ecoli, (g) wine, (h) ionosphere.

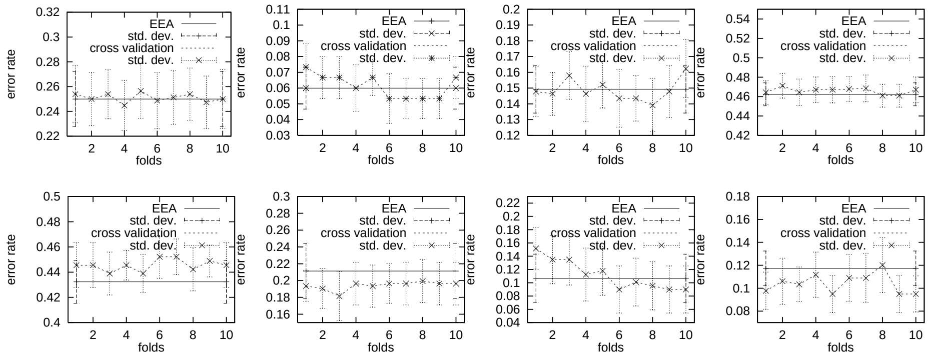

Fig. 3. Error rate (estimated by 10-fold cross validation) when either expected error analysis or n-fold cross validation is used to determine the number of leaf nodes. (a) diabetes, (b) iris, (c) crx, (d) cmc, (e) cleveland, (f) ecoli, (g) wine, (h) ionosphere.

6 Discussion and Related Work

Kearns and Mansour [8] proposed a nonparametric Chernoff-based pruning rule. Their rule removes all subtrees *unless* it can prove that the subtree really enhances the generalization performance. Therefore, the algorithm has a bias towards over-pruning the tree. Bayesian or MDL-based pruning strategies (*e.g.*, [11]) can also be seen as not being parametric. But they require additional information in terms of the prior probability of target densities ($p(p(x, y))$, in our formalism). In a way, this prior contains even more information than $\pi(\epsilon|H_i)$ because it tells something about all learning problems whereas $\pi(\epsilon|H_i)$ is a property of a certain given learning problem. Our experiments may slightly strengthen the belief (but do not prove) that exhaustive learning improves generalization slightly over greedy learning. (Relatively) fast exhaustive decision tree learners that are restricted to balanced trees have been presented in [1, 4]. Unfortunately, exhaustive learning is not feasible when the largest considered tree possesses many leaf nodes. However, our experiments show that a near optimal number of leaf nodes can be determined by means of our analysis as accurately as by 10-fold cross validation in $O(n \times i_{max} \times m)$ (while exhaustive tree learning is exponential in i_{max}). Our analysis predicts the generalization error rate of an exhaustive learner even when it would be far too expensive to actually run the learner. This opens the opportunity to determine the optimal number of leaf nodes very efficiently using our analysis and to invoke an exhaustive learner in case the optimal number of leaves is small. The speed-up that our regularization procedure achieves compared to n-fold cross validation is exponential when the underling learner is exhaustive, and is roughly a factor of n for greedy learners.

The analysis of the generalization error of the empirical error minimizing decision tree with i leaf nodes opens some new insights. Regularization algorithms that penalize the complexity of decision trees cannot directly minimize the generalization error $\epsilon(h_i^L)$ because empirical error and complexity do not suffice to infer the generalization error – therefore they possess a parameter that has to be adjusted for all problems (*e.g.,* [13, 20, 14]). However, we have seen that the missing information is contained in $\pi(\epsilon|H_i)$, a density that can often be estimated for a given i in $O(\log m)$ when a sample is given. Our analysis is an *actual-case* analysis (for a given learner and a given learning problem), rather than a (PAC-style) *worst-case* analysis (for the worst possible problem). Compared to earlier *actual case* analyses [17, 16, 5] our analysis is based on weaker assumptions. Compared to [18], our analysis is considerably simpler and, most importantly, covers greedy learners. An actual case analysis for Naive Bayesian classifiers that is guided by a similar idea has been presented by Langley and Sage [10], an actual case analysis for linear neural networks is given in [6].

ACKNOWLEDGMENT

Thanks to Ross Quinlan for an inspiring discussion at last year's machine learning conference. Thanks to Stefan Wrobel for helpful comments.

References

1. Peter Auer, Robert C. Holte, and Wolfgang Maass. Theory and applications of agnostic PAC-learning with small decision trees. In *12th ICML*, pages 21–29, 1995.
2. C. Blake and C. Merz. UCI repository of machine learning data sets. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1999.
3. L. Breiman, J Friedman, R. Ohlsen, and C. Stone. *Classification and Regression Trees.* Pacific Grove, 1984.
4. D. Dobkin, D. Gunopoulos, and S. Kasif. Computing optimal shallow decision trees. In *Proc. International Workshop on Mathematics in Artificial Intelligence*, 1996.
5. P. Domingos. Process-oriented estimation of generalization error. In *IJCAI-99*, 1999.
6. K. Fukumizu. Generalization error of linear neural networks in unidentifiable cases. In *Algorithmic Learning Theory*, 1999.
7. W. Gilks, S. Richardson, and D. Spiegelhalter, editors. *Markov Chain Monte Carlo in Practice.* Chapman & Hall, 1995.
8. M. Kearns and Y. Mansour. A fast, bottom-up decision tree pruning algorithm with near optimal generalization. In *ICML-98*, pages 269–277, 1998.
9. M. Kearns, Y. Mansour, A. Ng, and D. Ron. An experimental and theoretical comparison of model selection methods. *Machine Learning Journal*, 27:7–50, 1997.
10. P. Langley and S. Sage. Tractable average case analysis of naive bayes classifiers. In *ICML-99*, pages 220–228, 1999.
11. M. Mehta, J. Rissanen, and R. Agrawal. MDL-based decision tree pruning. In *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*, pages 216–221, August 1995.
12. N. Metropolis, A. Rosenbluth, M. Teller, and E. Teller. Equations of state calculations by fast computing machine. *J. Chem. Phys.*, 21:1087–1091, 1953.
13. J. Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3:319–342, 1989.

14. J. R. Quinlan. *C4.5 Programs for Machine Learning.* Morgan Kaufmann Publisher, 1993.
15. T. Scheffer. *Error Estimation and Model Selection.* Infix, 1999.
16. T. Scheffer and T. Joachims. Estimating the expected error of empirical minimizers for model selection. Technical Report TR 98-9, Technische Universitaet Berlin, 1998.
17. T. Scheffer and T. Joachims. Estimating the expected error of empirical minimizers for model selection (abstract). AAAI-98, 1998.
18. T. Scheffer and T. Joachims. Expected error analysis for model selection. In *ICML-99*, 1999.
19. G. Toussaint. Bibliography on estimation of misclassification. *IEEE Transactions on Information Theory IT20*, 4:472–479, 1974.
20. S. Weiss and N. Indurkhya. Small sample decision tree pruning. In *Proc. ICML-94*, pages 335–342, 1994.