Journal of Machine Learning Research 5 (2004) 1107–1126 Submitted 12/03, Revised 4/04; Published 9/04

Selective Rademacher Penalization and Reduced Error Pruning of Decision Trees*

Matti Kääriäinen MATTI.KAARIAINEN@CS.HELSINKI.FI
Tuomo Malinen
Department of Computer Science, University of Helsinki
P.O. Box 68
FI-00014 Helsinki, Finland
Tapio Elomaa ELOMAA@CS.TUT.FI
Institute of Software Systems, Tampere University of Technology
P.O. Box 553
FI-33101 Tampere, Finland

Editor: Peter L. Bartlett

Abstract

Rademacher penalization is a modern technique for obtaining data-dependent bounds on the generalization error of classifiers. It appears to be limited to relatively simple hypothesis classes because of computational complexity issues. In this paper we, nevertheless, apply Rademacher penalization to the in practice important hypothesis class of unrestricted decision trees by considering the prunings of a given decision tree rather than the tree growing phase. This study constitutes the first application of Rademacher penalization to hypothesis classes that have practical significance. We present two variations of the approach, one in which the hypothesis class consists of all prunings of the initial tree and another in which only the prunings that are accurate on growing data are taken into account. Moreover, we generalize the error-bounding approach from binary classification to multi-class situations. Our empirical experiments indicate that the proposed new bounds outperform distribution-independent bounds for decision tree prunings and provide non-trivial error estimates on real-world data sets.

Keywords: generalization error analysis, data-dependent generalization error bounds, Rademacher complexity, decision trees, reduced error pruning

1. Introduction

Data-dependent bounds on generalization error of classifiers are bridging the gap that has existed between theoretical and empirical results since the introduction of computational learning theory. They allow to take situation specific information into account, whereas distribution-independent results need to hold in all imaginable situations. Using *Rademacher complexity* (Koltchinskii, 2001; Bartlett and Mendelson, 2002) to bound the generalization error of a training error minimizing classifier is a fairly new approach that has not yet been tested in practice extensively.

Rademacher penalization is in principle a general method applicable to any hypothesis class. However, in practice it does not seem amenable to complex hypothesis classes because the standard

[*] This article is dedicated to the memory of the second author who unexpectedly passed away on June 6, 2004 at the age of twenty-seven.

method for computing Rademacher penalties relies on the existence of an empirical risk minimization algorithm for the hypothesis class in question. The first practical experiments with Rademacher penalization used real intervals as the hypothesis class (Lozano, 2000). Elomaa and Kääriäinen (2002) have applied Rademacher penalization to two-level decision trees, which can be learned efficiently in the agnostic PAC model (Auer et al., 1995).

General decision tree growing algorithms are necessarily heuristic because of the computational complexity of finding optimal decision trees (Grigni et al., 2000). Moreover, the hypothesis class consisting of unrestricted decision trees is so vast that traditional generalization error analysis techniques cannot provide non-trivial bounds for it. Nevertheless, top-down induction of decision trees by, e.g., C4.5 (Quinlan, 1993) produces results that are very competitive in prediction accuracy with better motivated approaches. We consider the usual two-phase process of decision tree learning; after growing a tree, it is pruned in order to reduce its dependency on the growing data and to better reflect characteristics of future data. Because of the practical success of decision tree learning, prunings of an induced decision tree can be considered an expressive class of hypotheses.

We apply Rademacher penalization to general decision trees by considering, not the tree growing phase, but rather the pruning phase. The idea is to view decision tree pruning as empirical risk minimization in the hypothesis class consisting of all prunings of an induced decision tree. First a heuristic tree growing procedure is applied to growing data to produce a decision tree. Then a pruning algorithm, for example the *reduced error pruning* (REP) algorithm of Quinlan (1987), is applied to the grown tree and a set of pruning data. As REP is known to be an efficient empirical risk minimization algorithm for the class of prunings of a decision tree (Elomaa and Kääriäinen, 2001), it can be used to compute the Rademacher penalty for this hypothesis class. Thus, by viewing decision tree pruning as empirical risk minimization in a data-dependent hypothesis class, we can bound the generalization error of prunings by Rademacher penalization. We also extend this generalization error analysis framework to the multi-class setting.

Standard Rademacher penalization still requires to take the whole hypothesis class into account. All possible prunings of the decision tree have to be evaluated. The prunings that evaluate best on randomly relabeled data—and, therefore, badly on the original data—essentially determine the error bound. However, in practice only prunings that have relatively small empirical error on the set of growing data are viable candidates for the final hypothesis. For this reason we restrict the pruning algorithm to operate on the much smaller class of hypotheses that consists of those prunings that make few mistakes on the set of growing data. To apply Rademacher penalization to this restricted class of hypotheses, we devise an empirical risk minimization algorithm for it. The new pruning algorithm, called k-REP, finds the most accurate pruning with respect to a set of pruning data among those prunings that make at most k mistakes on the set of growing data. The algorithm is based on dynamic programming and works in time cubic in the number of growing examples and linear in the number of pruning examples and the size of the decision tree to be pruned.

We evaluate the practical application potential of data-dependent error bounds empirically. Our experiments show that Rademacher penalization applied to prunings found by REP provides reasonable generalization error bounds on real-world data sets. The results for k-REP are even better. Although the bounds still overestimate the test set error, they are much tighter than distribution-independent bounds for prunings when the data sets are large.

This paper is organized as follows. In Section 2 we recapitulate the main idea of data-dependent generalization error analysis. We concentrate on Rademacher penalization, which we also extend to cover the multi-class case. Section 3 concerns pruning of decision trees, reduced error pruning of decision trees being the main focus. The k-REP algorithm together with a correctness proof and time complexity analysis is presented in Section 4. Combining Rademacher complexity calculation and decision tree pruning is the topic of Section 5. Empirical evaluation of the proposed approach is presented in Section 6 and, finally, Section 7 presents the concluding remarks of this study.

2. Rademacher Penalties

Let $S = \{\, (x_i, y_i) \mid i = 1, \ldots, n \,\}$ be a sample of n examples $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$ each of which is drawn independently from some unknown probability distribution on $\mathcal{X} \times \mathcal{Y}$. In the PAC and statistical learning settings one usually assumes that the learning algorithm chooses its hypothesis $h\colon \mathcal{X} \to \mathcal{Y}$ from some fixed hypothesis class $\mathcal{H}$. Under this assumption generalization error analysis provides theoretical results bounding the generalization error of hypotheses $h \in \mathcal{H}$ which is allowed to depend on the sample, the learning algorithm, and the properties of the hypothesis class. We consider the multi-class setting, where $\mathcal{Y}$ may contain more than two labels.

Let P be the unknown probability distribution according to which the examples are drawn. The *generalization error* of a hypothesis h is the probability that a randomly drawn example (x, y) is misclassified:

$$\varepsilon_P(h) = P(h(x) \neq y).$$

The general goal of learning, of course, is to find a hypothesis with a small generalization error. However, since the generalization error depends on P, it cannot be computed directly based on the sample alone. We can try to approximate the generalization error of h by its *training error* on n examples:

$$\hat{\varepsilon}_n(h) = \frac{1}{n} \sum_{i=1}^{n} \ell(h(x_i), y_i),$$

where ℓ is the $0/1$ loss function

$$\ell(y, y') = \begin{cases} 1, & \text{if } y \neq y'; \\ 0, & \text{otherwise.} \end{cases}$$

Empirical Risk Minimization (ERM) (Vapnik, 1982) is a principle that suggest choosing the hypothesis $h \in \mathcal{H}$ with minimal training error. In relatively small and simple hypothesis classes finding a minimum training error hypothesis is computationally feasible. To guarantee that ERM yields hypotheses with small generalization error, one can try to bound $\sup_{h \in \mathcal{H}} |\varepsilon_P(h) - \hat{\varepsilon}_n(h)|$. Under the assumption that the examples are independent and identically distributed (i.i.d.), whenever $\mathcal{H}$ is not too complex, the difference of the training error of the hypothesis h on n examples and its true generalization error converges to 0 in probability as n tends to infinity.

The most common approach to deriving generalization error bounds is based on the VC dimension of the hypothesis class (Vapnik and Chervonenkis, 1971; Blumer et al., 1989). The problem with this approach is that it provides optimal results only in the worst case—when the underlying probability distribution is as bad as it can be. Thus, the generalization error bounds based on VC dimension tend to be overly pessimistic. Moreover, the VC dimension bounds are hard to extend to the multi-class setting. Data-dependent generalization error bounds, on the other hand, can be provably almost optimal for any given domain (Koltchinskii, 2001). In the following we review the foundations of a recent promising approach to bounding the generalization error.

A *Rademacher random variable* takes values $+1$ and -1 with probability 1/2 each. Let $r_1, r_2, \ldots, r_n$ be a sequence of Rademacher random variables independent of each other and the data $(x_1,y_1), \ldots, (x_n,y_n)$. The *Rademacher penalty* of the hypothesis class $\mathcal{H}$ is defined as

$$R_n(\mathcal{H}) = \sup_{h \in \mathcal{H}} \left| \frac{1}{n} \sum_{i=1}^{n} r_i \ell(h(x_i), y_i) \right|.$$

Rademacher penalty is, thus, a random variable depending both on the random choice of the learning sample $(x_1,y_1), \ldots, (x_n,y_n)$ and on the randomness injected through the random variables $r_1, \ldots, r_n$. The following symmetrization inequality (Van der Vaart and Wellner, 2000), which also covers the multi-class setting, connects Rademacher penalties to generalization error analysis.

Theorem 1 *The inequality*

$$\mathbf{E}\left[\sup_{h \in \mathcal{H}} |\varepsilon_P(h) - \hat{\varepsilon}_n(h)|\right] \leq 2\mathbf{E}[R_n(\mathcal{H})]$$

holds for any distribution P, number of examples n, and hypothesis class $\mathcal{H}$.

The random variables $\sup_{h \in \mathcal{H}} |\varepsilon_P(h) - \hat{\varepsilon}_n(h)|$ and $R_n(\mathcal{H})$ are sharply concentrated around their expectations (Koltchinskii, 2001). The concentration results are based on the following McDiarmid's (1989) bounded difference inequality.

Lemma 2 (McDiarmid's inequality) *Let $Z_1, \ldots, Z_n$ be independent random variables taking their values in a set A. Let $f \colon A^n \to \mathbb{R}$ be a function such that over all $z_1, \ldots, z_n, z_i' \in A$*

$$\sup |f(z_1, \ldots, z_i, \ldots, z_n) - f(z_1, \ldots, z_i', \ldots, z_n)| \leq c_i$$

for some constants $c_1, \ldots, c_n \in \mathbb{R}$. Then for all $\varepsilon > 0$

$$\mathbf{P}(f(Z_1, \ldots, Z_n) - \mathbf{E}[f(Z_1, \ldots, Z_n)] \geq \varepsilon) \text{ and}$$
$$\mathbf{P}(\mathbf{E}[f(Z_1, \ldots, Z_n)] - f(Z_1, \ldots, Z_n) \geq \varepsilon)$$

are upper bounded by

$$\exp\left(-2\varepsilon^2 \Big/ \sum_{i=1}^{n} c_i^2\right).$$

Using McDiarmid's inequality one can bound the generalization error of hypotheses using their training error and Rademacher penalty as follows.

Lemma 3 *Let $h \in \mathcal{H}$ be arbitrary. Then with probability at least $1 - \delta$*

$$\varepsilon_P(h) \leq \hat{\varepsilon}_n(h) + 2R_n(\mathcal{H}) + 5\eta(\delta, n), \tag{1}$$

where $\eta(\delta, n) = \sqrt{\ln(2/\delta)/(2n)}$ is a hypothesis class independent error term that goes to zero as the number of examples increases.

Proof Observe that replacing a pair $((x_i,y_i),r_i)$ consisting of an example (x_i,y_i) and a Rademacher random variable r_i by any other pair $((x_i',y_i'),r_i')$ may change the value of $R_n(\mathcal{H})$ by at most $2/n$. Lemma 2 applied to the i.i.d. random variables $((x_1,y_1),r_1),\ldots,\ ((x_n,y_n),r_n)$ and the function $R_n(\mathcal{H})$ yields

$$\mathbf{P}(R_n(\mathcal{H}) \leq \mathbf{E}[R_n(\mathcal{H})] - 2\eta(\delta,n)) \leq \frac{\delta}{2}. \tag{2}$$

Similarly, changing the value of any example (x_i,y_i) can change the value of $\sup_{h\in\mathcal{H}} |\varepsilon_P(h) - \hat{\varepsilon}_n(h)|$ by no more than $1/n$. Thus, applying Lemma 2 again to $(x_1,y_1),\ldots,(x_n,y_n)$ and $\sup_{h\in\mathcal{H}} |\varepsilon_P(h) - \hat{\varepsilon}_n(h)|$ gives

$$\mathbf{P}\left(\sup_{h\in\mathcal{H}} |\varepsilon_P(h) - \hat{\varepsilon}_n(h)| \geq \mathbf{E}\left[\sup_{h\in\mathcal{H}} |\varepsilon_P(h) - \hat{\varepsilon}_n(h)|\right] + \eta(\delta,n)\right) \leq \frac{\delta}{2}. \tag{3}$$

To bound the generalization error of a hypothesis $g \in \mathcal{H}$ observe that

$$\varepsilon_P(g) \leq \hat{\varepsilon}_n(g) + \sup_{h\in\mathcal{H}} |\varepsilon_P(h) - \hat{\varepsilon}_n(h)|.$$

Hence, by inequality (3), with probability at least $1-\delta/2$

$$\begin{aligned} \varepsilon_P(g) &\leq \hat{\varepsilon}_n(g) + \mathbf{E}\left[\sup_{h\in\mathcal{H}} |\varepsilon_P(h) - \hat{\varepsilon}_n(h)|\right] + \eta(\delta,n) \\ &\leq \hat{\varepsilon}_n(g) + 2\mathbf{E}[R_n(\mathcal{H})] + \eta(\delta,n), \end{aligned}$$

where the second inequality follows from Theorem 1. Finally, applying inequality (2) yields that with probability at least $1-\delta$

$$\varepsilon_P(g) \leq \hat{\varepsilon}_n(g) + 2R_n(\mathcal{H}) + 5\eta(\delta,n).$$

■

The usefulness of inequality (1) stems from the fact that its right-hand side depends only on the training sample and the Rademacher random variables, but not on P directly. Hence, all the data that is needed to evaluate the generalization error bound is available to the learning algorithm. Furthermore, Koltchinskii (2001) has shown that in the two-class situation the Rademacher penalty can be computed by an empirical risk minimization algorithm applied to relabeled training data. We now extend this method to the multi-class setting.

The expression for $R_n(\mathcal{H})$ is first written as the maximum of two suprema in order to remove the absolute value inside the original supremum:

$$R_n(\mathcal{H}) = \max\left(\sup_{h\in\mathcal{H}}\left\{\pm\frac{1}{n}\sum_{i=1}^{n} r_i \ell(h(x_i),y_i)\right\}\right).$$

The sum inside the supremum with positive sign is maximized by the hypothesis h_1 that tries to correctly classify those and only those training examples (x_i,y_i) for which $r_i = -1$. To formalize this, we associate each class $y \in \mathcal{Y}$ with a complement class label $\bar{y}$ that represents the set of all classes but y. We denote the set of these complement classes by $\overline{\mathcal{Y}}$ and extend the domain of the loss function ℓ to cover pairs $(y,z) \in \mathcal{Y} \times \overline{\mathcal{Y}}$ by setting $\ell(y,z) = 1$ if $z = \overline{y}$ and 0 otherwise. Using this notation, h_1 is the hypothesis that minimizes the empirical error with respect to a newly labeled training set $\{\,(x_i,z_i)\,\}_{i=1}^n$, where

$$z_i = \begin{cases} y_i, & \text{if } r_i = -1; \\ \overline{y_i}, & \text{otherwise.} \end{cases}$$

The case for the supremum with negative sign is similar.

Altogether, the computation of the Rademacher penalty entails the following steps.

- Toss a fair coin n times to obtain a realization of the Rademacher random variable sequence $r_1,\ldots,r_n$.
- Change the label y_i to $\overline{y_i}$ if and only if $r_i = +1$ to obtain a new sequence of labels $z_1,\ldots,z_n$.
- Find functions $h_1,h_2 \in \mathcal{H}$ that minimize the empirical error with respect to the set of labels z_i and $\overline{z_i}$, respectively. Here, we follow the convention that $\overline{\overline{z}} = z$ for all $z \in \mathcal{Y} \cup \overline{\mathcal{Y}}$.
- Evaluate the Rademacher penalty given by the maximum of $|\{\,i : r_i = +1\,\}|/n - \hat{\varepsilon}(h_1)$ and $|\{\,i : r_i = -1\,\}|/n - \hat{\varepsilon}(h_2)$, where the empirical errors $\hat{\varepsilon}(h_1)$ and $\hat{\varepsilon}(h_2)$ are with respect to the labels z_i and $\overline{z_i}$, respectively.

In the two-class setting, the set $\overline{y}$ of all classes but y, $\mathcal{Y} \setminus \{\,y\,\}$, is a singleton. Thus, changing class y to $\overline{y}$ amounts to flipping the class label. It follows that a normal ERM algorithm can be used to find the hypotheses h_1 and h_2 and hence the Rademacher penalty can be computed efficiently provided that there exists an efficient ERM algorithm for the hypothesis class in question.

In the multi-class setting, however, a little more is required, since the sample on which the empirical risk minimization is performed may contain labels from $\overline{\mathcal{Y}}$ and the loss function differs from the standard 0/1-loss. This, however, is not a problem with the variants of REP covered in this paper nor with T2, a decision tree learning algorithm used in our earlier study, since all the algorithms can be easily adapted to handle this more general setting. The case for REP is covered in the next sections and for T2 in the paper by Auer et al. (1995).

3. Growing and Pruning Decision Trees

A decision tree (Breiman et al., 1984) is a rooted tree in which the inner nodes are equipped with *branching functions* and the leaves are labeled with classes. A branching function routes examples reaching a node to its children, thus defining for each example a unique root-leaf path. The classification of an example is determined by the label of the leaf to which the example is routed.

A common approach in top-down induction of decision trees is to first grow a tree that fits the training data well and, then, prune it to reflect less the peculiarities of the training data; i.e., to generalize better. Here, pruning means replacing some inner nodes of the tree with leaves and removing the parts of the tree that become unreachable from the root. Many heuristic approaches (Quinlan, 1987; Mingers, 1989; Esposito et al., 1997) as well as more analytical ones (Mansour, 1997; Kearns and Mansour, 1998) to pruning have been proposed. A special class of pruning algorithms are the on-line ones (Helmbold and Schapire, 1997; Pereira and Singer, 1999). Even these algorithms work by the two-phase approach: An initial decision tree is fitted to the data and its prunings are then used as experts that collectively predict the class of observed instances.

Reduced error pruning was originally proposed by Quinlan (1987). It produces an optimal pruning of a given tree—the smallest tree among those with minimal error with respect to a given set of *pruning examples* (Esposito et al., 1997; Elomaa and Kääriäinen, 2001). The REP algorithm works in two phases: First the set of pruning examples S is classified using the given tree T to be pruned. Counters that keep track of the number of examples of each class passing through each node are updated simultaneously. In the second phase—a bottom-up pruning phase—those parts of the tree that can be removed without increasing the error of the remaining hypothesis are pruned away. The pruning decisions are based on the node statistics calculated in the top-down classification phase.

REP can be viewed as an ERM algorithm for the hypothesis class consisting of all prunings of a given decision tree. Thus, it can be used to efficiently compute Rademacher penalties and, hence, also generalization error bounds for the class of prunings of a decision tree. This leads us to the following strategy. First, we use a standard heuristic decision tree induction algorithm to grow a C4.5-type decision tree based on a set of growing examples. The tree serves as a representation of the data-dependent hypothesis class that consists of its prunings. As C4.5 usually performs quite well on real-world domains, it is reasonable to assume—even though it cannot be proved—that the class of prunings contains some good hypotheses.

Having grown a decision tree, we use a separate pruning data set to select one of the prunings of the grown tree as our final hypothesis. In this paper, we use REP as our pruning algorithm, but in principle any other pruning algorithm using the same basic pruning operation could be used instead. However, since REP is an empirical risk minimization algorithm, the derived error bounds will be the tightest when combined with the prunings produced by REP.

3.1 Reducing the Number of Prunings

As argued above, the set of prunings of a decision tree is likely to contain accurate hypotheses. Still, most of the prunings—the ones performing badly on the growing set—are likely to be very inaccurate on the pruning data. If the growing and the pruning data sets resemble each other to any extent, which is a necessary condition for the two-phase learning paradigm to make sense in the first place, the pruning algorithm will not select any of these hypotheses with very bad performance on the set of growing data. Keeping these inaccurate prunings as part of the hypothesis class only makes the hypothesis class more complex and, hence, increases the Rademacher penalty associated with it.

Following the line of thought above, it would seem reasonable to restrict the pruning algorithm to select the final pruning from among those hypotheses that are relatively accurate on the set of growing data. In Section 4 we present in detail the k-REP pruning algorithm, which does exactly this by solving the following problem: given a decision tree and sets of growing and pruning data, find the most accurate pruning (w.r.t. the pruning data) of the tree among those prunings that make at most k mistakes on the growing data. The restriction to prunings that are accurate on the growing data adds to the combinatorial complexity of the search problem, but we are still able to solve the problem in cubic time by using dynamic programming. k-REP is an efficient ERM algorithm for the restricted class of prunings. Thus, it can be used to evaluate generalization error bounds based on Rademacher penalties in the same way as REP can be used in connection with the class of all prunings (Kääriäinen and Elomaa, 2003). Since k-REP operates on a subclass of the class of all prunings, the Rademacher penalties are in this case smaller.

In order to use k-REP one has to devise some strategy of choosing a value for k, that is, to define exactly what it means for a hypothesis to be accurate on a set of growing data. If k is very large, k-REP boils down to standard REP since a loose bound on mistakes does not rule any of the prunings out. On the other hand, too small a k may shrink the hypothesis class too small or even empty if none of the prunings meets the strict accuracy requirement. A theoretically well-motivated solution would be to consider all values of k and employ standard model selection techniques to pick the one that gives the best error bounds. However, the model selection phase would loosen the bounds as the confidence parameter δ would have to be split among the different values of k. Hence, the best bound obtainable using model selection would unavoidably be larger than the best bound achievable if one could somehow pick a single fortunate choice for k.

In practice, the number of errors the original decision tree makes on the set of growing data is a good baseline to which the accuracy of the prunings can be related—we want the prunings considered by k-REP to be almost as accurate on the growing data as the original decision tree. Thus, we will select k to be some constant factor $c > 1$ times the number of errors the original tree makes on the growing data. This way of choosing the value of k is, of course, just an intuitively motivated heuristic, but so is the whole decision tree growing procedure that determines the original class of prunings in the first place. Our empirical experiments show this strategy works well on real world data sets.

A similar idea to the one behind k-REP is employed in the *shell decomposition bounds* of Langford and McAllester (2000), who show that the effective complexity of a hypothesis class can be measured by the complexity of the sub-class (or shell of hypothesis) that consists of only the almost most accurate hypotheses of the original class. The shells, however, are defined based on the same data that is used for selecting the final hypothesis, whereas in the case of k-REP the sub-class of accurate hypotheses is selected based on the growing data and the final hypothesis is chosen based on the pruning data. Also local Rademacher complexities (Bartlett et al., 2002, 2004; Lugosi and Wegkamp, 2004) and other local complexity measures (Koltchinskii and Panchenko, 2000; Massart, 2000; Mendelson and Philips, 2003) aim at taking into account only those parts of the model that are relevant for the given learning task. However, these methods have not been tested in practice as evaluating the local complexity measures involves some computational and other practical problems that have not been attacked yet.

3.2 Related Pruning Algorithms

REP produces the smallest of the most accurate prunings of a given decision tree, where accuracy is measured with respect to the pruning set. Other approaches for producing optimal prunings for different optimality criteria have also been proposed (Breiman et al., 1984; Bohanec and Bratko, 1994; Oliver and Hand, 1995; Almuallim, 1996). These criteria typically take both the size of the resulting pruning and its accuracy on growing data into account. As pruning tends to reduce growing set accuracy, one typically has to make a compromise between maintaining the initial growing set accuracy and finding a small pruning. For example, Bohanec and Bratko (1994) as well as Almuallim (1996) have studied how to efficiently find the smallest pruning that satisfies a given minimum accuracy requirement.

The strategy of using one data set for growing a decision tree and another for pruning it closely resembles the on-line pruning setting (Helmbold and Schapire, 1997; Pereira and Singer, 1999). In it the prunings of the initial decision tree are viewed as a pool of experts. Thus, pruning is performed on-line, while giving predictions to new examples, rather than in a separate pruning phase. The main advantage of these on-line methods is that no statistical assumptions about the data generating process are needed and still the combined prediction and pruning strategy can be proved to be competitive with the best possible pruning of the initial tree. However, these approaches do not choose or maintain one pruning of the given decision tree, but rather a weighted combination of prunings, which may be impossible to interpret by human experts. Also, the loss bounds are meaningful only for very large data sets and there exists no empirical evaluation of the performance of the on-line pruning methods.

The pruning algorithms of Mansour (1997) and Kearns and Mansour (1998) are very similar to REP in spirit. The main difference with these algorithms and REP is the fact that they do not require the sample S on which pruning is based to be independent of the tree T; i.e., T may well have been grown based on S. Moreover, the pruning criterion in both methods is a kind of a *cost-complexity* condition (Breiman et al., 1984) that takes both the observed classification error and (sub)tree complexity into account. Both algorithms are *pessimistic*: They try to bound the true error of a (sub)tree by its training error. Since the training error is by nature optimistic, the pruning criterion has to compensate it by being pessimistic about the error approximation.

Both Mansour (1997) and Kearns and Mansour (1998) provide generalization error analyses for their algorithms. The bound presented in (Mansour, 1997) measures the complexity of the class of prunings by the size of the tree to be pruned. If this size or an upper bound for it is known in advance, the bound applies also when the pruning data is not independent of the tree to be pruned. Kearns and Mansour (1998) prove that the generalization error of the pruning produced by their algorithm is bounded by that of the best pruning of the given tree plus a complexity penalty. However, the penalty term can grow intolerably large and cannot be evaluated because of its dependence on the unknown optimal pruning and hidden constants.

One shortcoming of the two-phase decision tree induction approach is that there does not exist any well-founded approach for deciding how much data to use for the training and pruning phases. Only heuristic data set partitioning schemes are available. However, the simple rule of using, e.g., two thirds of the data for growing and the rest for pruning has been observed to work well in practice (Esposito et al., 1997). If the initial data set is very large, it may be computationally infeasible to use all the data for growing or pruning. In that case one can use heuristic sequential sampling methods for selecting the size of the growing set and determine the size of the pruning set, e.g., by using progressive Rademacher sampling (Elomaa and Kääriäinen, 2002). Because REP is an efficient linear-time algorithm, it is not hit hard by overestimated pruning sample size.

4. k-Optimal REP Prunings

Given a decision tree to be pruned and a set of pruning examples, REP finds the pruning that minimizes error on the pruning set; no consideration is given to the growing set error of the resulting hypothesis. In Section 3.1, we motivated the idea of imposing a restriction also on the growing set error of REP prunings. Clearly, in order to be able to prune at all, one has to give up some accuracy on the data that was used to grow the tree. This naturally leads to the idea of finding REP prunings with growing set error at most some threshold value k.

Let T be a (subtree of a) decision tree, $\hat{\varepsilon}_g(T)$ its growing set error, $\hat{\varepsilon}_p(T)$ its pruning set error, and $|T|$ its size. Let $\mathcal{P}(T)$ be the set of all the prunings of T.

Definition 4 *A k-*optimal REP pruning *of a decision tree T is a $T' \in \mathcal{P}(T)$ that has $\hat{\varepsilon}_g(T') \leq k$, if one exists, and for which*

- $\hat{\varepsilon}_p(T') = \min\{\, \hat{\varepsilon}_p(T'') \mid T'' \in \mathcal{P}(T),\ \hat{\varepsilon}_g(T'') \leq k \,\}$*, and*
- $|T'| = \min\{\, |T''| \mid T'' \in \mathcal{P}(T),\ \hat{\varepsilon}_p(T'') = \hat{\varepsilon}_p(T') \,\}$*,*

If there is no $T' \in \mathcal{P}(T)$ satisfying the criteria, k-optimal REP *pruning of T is undefined.*

For clarity, we consider only binary trees at first. Let T be a decision tree with root node N. Assume that, for each i, $0 \leq i \leq k$, we know i-optimal REP prunings of the children T_1 and T_2 of the root node N of T. Denote these by $T_1^0, \ldots, T_1^k$ and $T_2^0, \ldots, T_2^k$, respectively. Choosing any pair (T_1^u, T_2^v) of these prunings defines a pruning of T in the obvious way; let $\langle N, T_1^u, T_2^v \rangle$ denote this pruning.

In this paper we assume that leaf labels for decision tree prunings are determined by the growing data. Alternative leaf labeling strategies are discussed by Elomaa and Kääriäinen (2001) and a k-REP pruning algorithm resembling the one presented next could be derived for these labeling strategies as well. Let N_g denote the single-leaf pruning of T, i.e., a leaf labeled with the majority class of growing examples reaching T. The following result suggests a dynamic programming technique for finding k-optimal REP prunings, which is described subsequently.

Theorem 5 *If the k-optimal* REP *pruning of a decision tree T is defined, it is either the leaf N_g or of the form $\langle N, T_1^u, T_2^v \rangle$, where $u + v = k$ and T_1^u and T_2^v are u- and v-optimal* REP *prunings of the left and the right subtree of T, respectively.*

Proof Let T' be the k-optimal REP pruning of a decision tree T. If T' is N_g, then we have the claim. Otherwise, T' consists of a root node N and two subtrees T_1' and T_2', which respectively are prunings of the subtrees T_1 and T_2 of T. Now, $\hat{\varepsilon}_g(T') = \hat{\varepsilon}_g(T_1') + \hat{\varepsilon}_g(T_2') \leq k$, which means that there must exist u and v such that $u + v = k$, $\hat{\varepsilon}_g(T_1') \leq u$ and $\hat{\varepsilon}_g(T_2') \leq v$.

Let T_1^u be a u-optimal REP pruning of T_1 and assume that T_1' is not. By Definition 4 either $\hat{\varepsilon}_p(T_1') > \hat{\varepsilon}_p(T_1^u)$ or $|T_1'| > |T_1^u|$. Both cases contradict the k-optimality of pruning T', because the tree $\langle N, T_1^u, T_2' \rangle$ would be better than it. If $\hat{\varepsilon}_p(T_1') > \hat{\varepsilon}_p(T_1^u)$, then

$$\hat{\varepsilon}_p(T') = \hat{\varepsilon}_p(T_1') + \hat{\varepsilon}_p(T_2') > \hat{\varepsilon}_p(T_1^u) + \hat{\varepsilon}_p(T_2') = \hat{\varepsilon}_p(\langle N, T_1^u, T_2' \rangle).$$

If, on the other hand, $|T_1'| > |T_1^u|$, then

$$|T'| = |T_1'| + |T_2'| + 1 > |T_1^u| + |T_2'| + 1 = |\langle N, T_1^u, T_2' \rangle|.$$

Therefore, T_1' has to be a u-optimal REP pruning of T_1. Similar argumentation also proves the v-optimality of T_2'. ■

What Theorem 5 effectively says is that the k-optimal REP pruning of a tree T is either N_g or a combination of u- and v-optimal REP prunings of the children of its root node for some u and v summing up to k. Therefore, by going through each of the mentioned prunings, and minimizing over them first by pruning error, then by size, we can find k-optimal REP prunings of T. The k-optimal REP prunings are easy to find for trees consisting of single leafs. Combining this with a bottom

Algorithm 6 *Find k-optimal* REP *prunings.*

```
1   for each $i \in \{0,\ldots,\min(n,k)\}$ do
2       $\hat{\varepsilon}_p(T^i) \leftarrow \infty$;
3       $|T^i| \leftarrow \infty$
4   od;
5   if $N$ is not a leaf then
6       for each $i \in \{0,\ldots,\min(n,k)\}$ do
7           for each $(u,v)$ such that $u+v=i$ do
8               $T' \leftarrow \langle N, T_1^u, T_2^v \rangle$;
9               if $\hat{\varepsilon}_p(T') < \hat{\varepsilon}_p(T^i)$ then $T^i \leftarrow T'$ fi;
10              else if $\hat{\varepsilon}_p(T') = \hat{\varepsilon}_p(T^i)$ and $|T'| < |T^i|$ then $T^i \leftarrow T'$ fi
11          od
12      od
13  fi;
14  for each $i$, $i \in \{\hat{\varepsilon}_g(N_g),\ldots,\min(n,k)\}$ do
15      if $\hat{\varepsilon}_p(N_g) \leq \hat{\varepsilon}_p(T^i)$ then $T^i \leftarrow N_g$ fi
16  od;
```

up sweep of T yields a dynamic programming technique for the task at hand. The step of dynamic programming is given as Algorithm 6, which finds T^i for each i, $0 \leq i \leq \min(n,k)$, where n is the number of growing examples that reach the node. T^i is undefined for any i for which $|T^i| = \infty$ after running the algorithm.

The generalization of Theorem 5 (and Algorithm 6) to non-binary trees is straightforward. For a t-way split, one has to go through all the partitions of each i, $0 \leq i \leq \min(n,k)$, into t addends. This makes the time complexity exponential in the number of branches in the split, as the number of such partitions grows exponentially in t.

Let us consider the time complexity of Algorithm 6. Clearly, the loop on lines 14–16 works in time linear in $\min(n,k)$. In the loop on lines 6–12, one has to check i partitions for each i, $0 \leq i \leq \min(n,k)$. This makes the time complexity of processing a single node with a binary split $O(\min(n,k)^2)$, where n is the number of growing examples that reach the node.

Now consider a binary tree grown on n examples. First note that at most n growing examples reach the nodes of any particular level of the tree. Consider an arbitrary level with $w \leq n$ nodes, with $n_1,\ldots,n_w$ growing examples reaching them. By the above bound for a single node, the computation on the level takes $O(\sum_{i=1}^{w} \min(n_i,k)^2)$ steps. Now, it is clear that $\sum_{i=1}^{w} \min(n_i,k) \leq n$ holds, and this implies $\sum_{i=1}^{w} \min(n_i,k)^2 \leq n^2$, so $O(n^2)$ is an upper bound for the time complexity on any single level of the tree. A tree grown on n examples has at most n levels, which makes the worst case complexity $O(n^3)$.

The above result assumes that the pruning errors on lines 9, 10, and 15 can be evaluated in constant time. This can be achieved by equipping the nodes of the original tree with counters telling the class frequencies of pruning examples going through them. Initializing such counters can be done in time linear in the number of pruning examples and the size of the tree to be pruned. As the algorithm does not need to access the pruning data after this preprocessing step, the time complexity with respect to the amount of pruning data is linear.

The $O(n^3)$ time complexity result can be strengthened if we make more assumptions on the decision tree to be pruned or the distribution of the growing examples to the tree. For example, if the depth of the tree can be assumed to be $O(\log n)$, the upper bound on the time complexity of k-REP is reduced to $O(n^2 \log n)$. As another special case, assume that the set of growing examples is halved in each node of a tree with n leaves. Then, the time complexity reduces to

$$c \sum_{i=0}^{\log n} 2^i \cdot \left(\min\left(\frac{n}{2^i}, k\right)\right)^2 = O(n^2),$$

where each addend corresponds to a single level of the tree.

5. Combining Rademacher Penalization and Decision Tree Pruning

When using REP or k-REP, the data sets used in growing the tree and pruning it are independent of each other. Therefore, any standard generalization error analysis technique can be applied to the resulting pruning as if the hypothesis class from which the pruning was selected was fixed in advance. A formal argument justifying this would be to carry out the generalization error analysis conditioned on the training data and then to argue that the bounds hold unconditionally by taking expectations over the selection of the training data set.

By the above argument, the theory of Rademacher penalization can be applied to the data-dependent class of prunings. Therefore, we can use the results presented in Section 2 to provide generalization error bounds for prunings found by REP, k-REP, or any other pruning algorithm. Moreover, since both REP and k-REP are efficient ERM algorithms (linear and cubic time, respectively) for the related classes of prunings, the generalization error bounds can be evaluated efficiently.

To summarize, we propose the following decision tree learning strategy that provides a generalization error bound for the hypothesis it produces:

1. Split the available data into a growing set and a pruning set.

2. Use, e.g., C4.5 (without pruning) on the growing set to induce a decision tree.

3. Find the smallest most accurate pruning of the tree built in the previous step using REP (or any other pruning algorithm) on the pruning set. This is the final hypothesis. Alternatively, choose a suitable k and use k-REP to find the most accurate pruning from the class of prunings making at most k errors on the set of growing data.

4. Evaluate the error bound as explained in Section 2 by running REP two more times. In case k-REP was used in step 3, use k-REP in place of REP here, too.

Even though the tree growing process is heuristic, the generalization error bounds for the prunings are provably true under the i.i.d. assumption. They are valid even if the tree growing heuristic fails, that is, when none of the prunings of the grown tree generalize well. In that case the bounds are, of course, unavoidably large. The situation is similar to, e.g., margin-based generalization error analysis (Cristianini and Shawe-Taylor, 2000), where the error bounds are good provided that the training data generating distribution is such that a hypothesis with a good margin distribution can be found. In our case the error bounds are tight provided that C4.5 produces a decision tree that has good prunings and is still relatively small so that the Rademacher penalty for the class of its prunings does not blow up. A good choice of k may help in keeping the penalty term in control, a situation resembling choosing the marginal parameter in margin-based generalization error analysis. The existing empirical evidence overwhelmingly demonstrates that C4.5 usually fares quite well, and our experiments presented in Section 6 indicate that a good choice of k really results in a notable decrease in the complexity term on real world data sets.

The value of k should ideally be so large that the hypothesis class associated with it includes the most accurate pruning w.r.t. pruning data and, at the same time, as small as possible to limit the complexity of the remaining hypothesis class to a minimum. This trade-off is hard to solve in general, since the decision on which k to choose has to be done prior to seeing the set of pruning data. In the following we will choose k to be some $c > 1$ times the number of errors the original decision tree makes on the set of growing data. This way we take into account the fact that the original tree most likely overfits the growing data set and thus has a smaller error than can be expected from prunings with good generalization. The empirical experiments indicate that $c = 1.1$ is a reasonable choice for all data sets we experimented with.

Generalization error bounds can be roughly divided into two categories: Those based on a training set only and those requiring a separate test set (Langford, 2002). Our generalization error bounds for prunings may be seen to lie somewhere between these two extremes, the bound for k-REP being the one closer to test set bounds. We use only part of the data in the tree growing phase that determines our hypothesis class. The rest—the set of pruning data—is used only for selecting a pruning and evaluating the generalization error bound. Thus, some of the information contained in the pruning set may be lost as it cannot be used in the tree induction phase. However, the pruning set is still used for the non-trivial task of selecting a good pruning, so that some of the information contained in it can be exploited in the final hypothesis. The pruning set is thus used as a test set for the outcome of the tree growing phase and also as a proper learning set in the pruning phase.

6. Empirical Evaluation

Before reporting and discussing the results obtained in our tests, we describe the distribution-independent bound used as comparison point to Rademacher penalization and briefly outline other aspects of the test setting.

6.1 Test Setting for Performance Comparison

The obvious performance reference for Rademacher penalization over decision tree prunings is to compare it to existing generalization error bounds. The bound of Kearns and Mansour (1998) is impossible to evaluate in practice because it requires knowing the depth and size of the pruning with the best generalization error. The bound presented by Mansour (1997) only requires knowing the maximum size of prunings in advance and would, thus, be applicable in our setting. However, Mansour's bound is clearly inferior to the simpler Occam's Razor type of bound to be introduced next and will, hence, be excluded from the empirical comparison. Bounds developed in the on-line pruning setting (Helmbold and Schapire, 1997) are incomparable with the one presented in this paper because of the different learning model. Thus, they will not be considered here.

The simplest—and as it turned out in our experiments, the tightest—existing generalization error bound which the Rademacher bound can be compared to is to our knowledge an Occam's Razor bound (Blumer et al., 1987; Langford, 2003) that is obtained by assigning equal-length codes to all prunings of the original decision tree. Equivalently, we assign equal prior probability to all prunings of the original tree. Since the leaf labels of the prunings are determined by the growing data, all that needs to be encoded is the set of those inner nodes that are to be replaced by leaves. A simple way to do so is to assign a bit for each of the $(d-1)/2$ inner nodes of a d node tree telling whether the node is pruned or not.

The simplistic code outlined above contains some redundancy as, e.g., the pruning consisting of a single leaf is represented by $2^{(d-1)/2-1}$ different codewords. However, it is easy to see that a binary tree with d nodes can have at least $2^{d/4}$ prunings; consider, e.g., the prunings obtainable from a balanced tree by pruning a subset of inner nodes next to the leaves. Thus, no less than $d/4$ bits will suffice if nothing but the size of the tree to be pruned is taken into account. To find out the optimal uniform code length given the whole tree to be pruned as a parameter, one would essentially have to count the number of prunings of the tree. We are not aware of an efficient algorithm for this task. On the other hand, using a non-uniform code length would introduce a bias to the bound that is not present in our proposed bounds. Thus, in our experiments we will use the code length approximation $d/4$, giving worst-case optimistic error bounds. Plugging this into the Chernoff Occam's Razor bound (Langford, 2003) we get that with probability at least $1-\delta$,

$$\varepsilon_P(h) < \hat{\varepsilon}_n(h) + \sqrt{\frac{\ln 2 \cdot d/4 + \ln(1/\delta)}{2n}},$$

where d is the number of the nodes of the tree and n is the size of the pruning set. This bound could be further improved by using the exact Occam's razor bound (Langford, 2003) instead, but we have not tried how significant the improvement would be. Note that this bound is data independent in the sense that the pruning data is taken into account only through the pruning error $\hat{\varepsilon}_n(h)$.

The error bounds based on Rademacher penalization depend on the data distribution so that their true performance can be evaluated only empirically. In our experiments we grow binary decision trees using a C4.5-type decision tree algorithm distributed in the Weka package (Witten and Frank, 1999). As a benchmark we use 15 data sets from the UCI Machine Learning Repository (Blake and Merz, 1998). In each experiment we allocate 10 percent of the data for testing and split the rest to growing and pruning sets. As the split ratio we chose 2:1 as suggested by Esposito et al. (1997). For the generated data set LED, we use 300,000 instances with 10 percent attribute noise. For k-REP we choose $c = 1.1$, i.e., k is 1.1 times the training error of the unpruned tree.

6.2 Empirical Observations

Table 1 and Figure 1 summarize the results over 10 random splits of the data sets. In Table 1 we present the decision tree sizes before and after pruning with k-REP and REP. Observe that the unpruned decision trees are very large, which means that the class of prunings may potentially be very complex. The results indicate that REP manages to decrease the tree sizes considerably. The sizes of k-REP prunings fall in many cases roughly halfway between the unpruned tree size and the size of the REP pruned tree.

Figure 1 presents the test set accuracies and error bounds based on Rademacher penalization and Occam's Razor. In all bounds, we set $\delta = 0.01$. Even though both bounds based on Rademacher

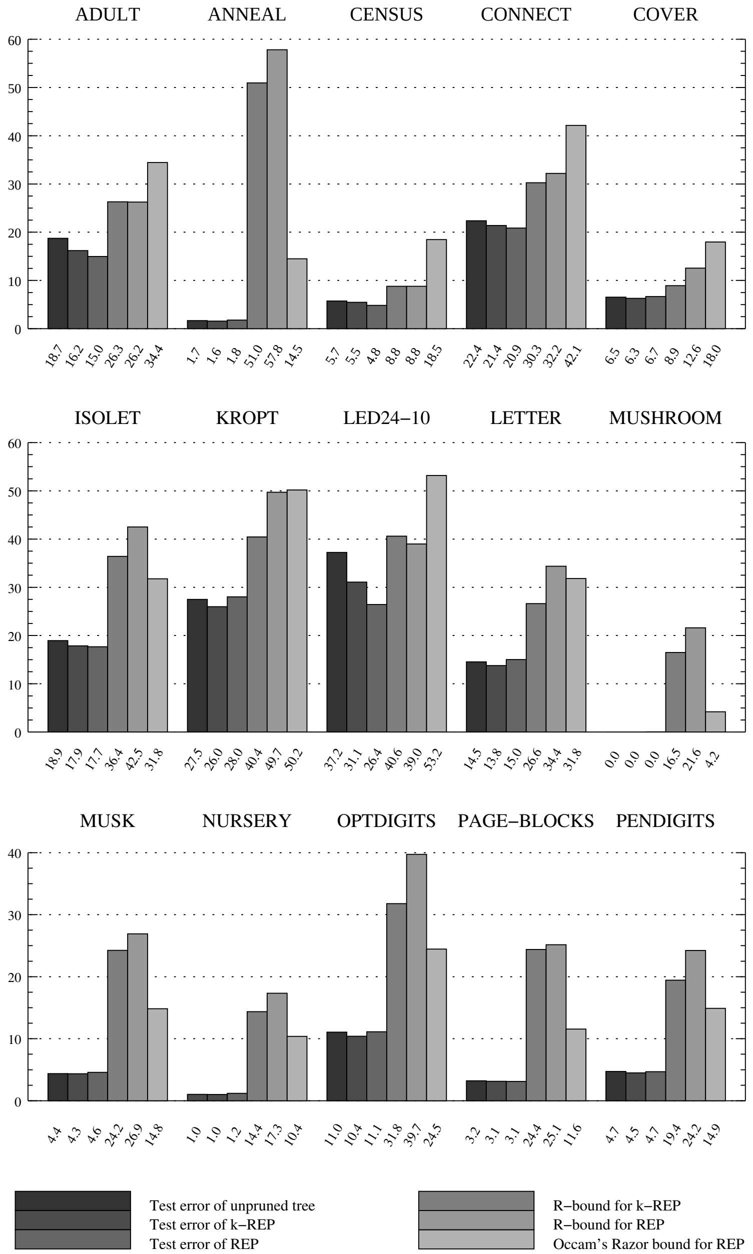

Figure 1: Averages of error bounds over 10 random splits of the data sets.

DATA SET	UNPRUNED	*k*-REP	REP
ADULT	7,507.6	3,898.6	1,600.6
ANNEAL	32.0	24.8	20.8
CENSUS	20,513.4	12,378.6	4,819.4
CONNECT	13,953.8	8,583.8	4,289.0
COVER	31,483.6	25,374.0	18,396.4
ISOLET	664.8	517.6	272.0
KROPT	7,317.4	5,328.8	3,572.4
LED24-10	90,564.8	43,689.4	9,041.6
LETTER	2,543.8	1,907.0	1,292.4
MUSHROOM	22.8	22.8	22.0
MUSK	224.8	186.0	120.0
NURSERY	392.0	349.4	306.8
OPTDIGITS	410.4	319.8	222.2
PAGE-BLOCKS	123.2	85.6	42.6
PEN-DIGITS	411.0	324.0	245.8

Table 1: Average sizes of trees over 10 random splits of the data sets.

penalization clearly overshoot the test set accuracies, they still provide reasonable estimates in many cases. Note that in the multi-class settings even error bounds above 50 percent are non-trivial.

Both bounding methods, the one based on Rademacher penalties and the one based on Occam's Razor, outperform the other on a number of data sets; there seems to be no clear overall winner. Notably, in many cases the difference between the better and worse method is quite large. On large data sets, the Rademacher bounds are consistently better; the converse holds for the small sets. The small amount of data blows up the hypothesis class independent term $\eta(\delta, n)$ to the extent that it starts to dominate the actual Rademacher penalty. The Occam's Razor bound is clearly better when the unpruned tree is small, since this situation keeps the penalty term related to it under control.

Rademacher bounds for *k*-REP turn out to be better than the REP bound in most cases. The only notable exception is the LED domain, where the pruning error of the best pruning is significantly lower than that of the best restricted pruning, while the Rademacher penalties for both classes are almost the same. In CENSUS INCOME the decrease of pruning error and growth of the Rademacher penalty cancel each other out so that the bounds for REP and *k*-REP are nearly equal.

We also conducted a set of experiments in order to see how the bound behaves as a function of c. The results indicate that decreasing c typically yields tighter bounds, but at the same time the actual quality of the prunings obtained deteriorates as c gets closer to 1. In the limiting case $c = 1$ there is no room left for pruning, so this extreme case effectively coincides with using the pruning set as a set of test data. Increasing c relaxes the restrictions on the pruning decisions and enables *k*-REP to find prunings with better empirical performance. The trade-off here is a special case of the fact that test error bounds are typically the tightest in practice even though using all the data in learning might yield a hypothesis with better generalization error. Our choice of $c = 1.1$ seems to be a good compromise between the tightness of the bound and the actual generalization performance of the obtained pruning.

The relative test performance of *k*-REP and REP is varied and neither method seems to be a clear winner. As *k*-REP produces larger prunings and is computationally more demanding than REP, there seems to be little motivation for using k-REP independently as a pruning method if error guarantees are not called for.

Our intention has been to carry out a feasibility study of the new technique of Rademacher penalization, rather than to aim at generalization error bounds directly applicable in the real world. However, the bounds that were obtained on larger data sets are sometimes tighter than one could have expected in advance. In the best cases the theoretical bounds already approach usability as performance guarantees of practical algorithms. Even though even the best of the proposed bounds always overestimates the test error, it is never totally unrealistic. Thus, we have demonstrated that Rademacher penalization represents a step toward the use of well-founded training set bounds in practical applications. Though, at the same time it is, unfortunately, not possible to draw too far-reaching positive conclusions from this study, because in the worst cases Rademacher penalization fails to deliver usable bounds and does not fare as well as the Occam's Razor bound on smaller data sets.

7. Conclusion

Modern generalization error bounding techniques that take the observed data distribution into account give far more realistic sample complexities and generalization error approximations than the distribution-independent methods. We have shown how one of these techniques, namely Rademacher penalization, can be applied to bound the generalization error of decision tree prunings, also in the multi-class setting. According to our empirical experiments the proposed theoretical bounds are often tighter than distribution-independent generalization error bounds for decision tree prunings. However, the new bounds still appear unable to faithfully describe the performance attained in practice.

As future work, we intend to carry out more thorough empirical experiments on the proposed methods. Also, we will look for better motivated ways of tuning the value of c and of determining the proportion of learning data allocated for pruning purposes. It would also be interesting to extend the two-phase generalization error analysis approach introduced here to other hypothesis classes, too.

Acknowledgments

We thank the anonymous referees for suggestions that improved the final version of this paper; in particular, for pointing out the possibility of using the Occam's Razor bound as the comparison point in our empirical evaluation.

References

Hussein Almuallim. An efficient algorithm for optimal pruning of decision trees. *Artificial Intelligence*, 83(2):347–362, 1996.

Peter Auer, Robert C. Holte, and Wolfgang Maass. Theory and application of agnostic PAC-learning with small decision trees. In Armand Prieditis and Stuart Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning*, pages 21–29, San Francisco, CA, 1995. Morgan Kaufmann.

Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Localized Rademacher complexities. In Jyrki Kivinen and Robert H. Sloan, editors, *Computational Learning Theory, Proceedings of the Fifteenth Annual Conference*, volume 2375 of *Lecture Notes in Artificial Intelligence*, pages 44–58, Berlin Heidelberg New York, 2002. Springer.

Peter L. Bartlett, Olivier Bousquet, and Shahar Mendelson. Local Rademacher complexities. *The Annals of Statistics*, 2004. To appear.

Peter L. Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Cathrine L. Blake and Christopher J. Merz. *UCI Repository of Machine Learning Databases*. University of California, Department of Information and Computer Science, Irvine, 1998. http://www.ics.uci.edu/∼mlearn/MLRepository.html.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, 1987.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

Marco Bohanec and Ivan Bratko. Trading accuracy for simplicity in decision trees. *Machine Learning*, 15(3):223–250, 1994.

Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, Pacific Grove, 1984.

Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

Tapio Elomaa and Matti Kääriäinen. An analysis of reduced error pruning. *Journal of Artificial Intelligence Research*, 15:163–187, 2001.

Tapio Elomaa and Matti Kääriäinen. Progressive Rademacher sampling. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence*, pages 140–145, Cambridge, MA, 2002. MIT Press.

Floriana Esposito, Donato Malerba, and Giovanni Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 (5):476–491, 1997.

Michelangelo Grigni, Vincent Mirelli, and Christos H. Papadimitriou. On the difficulty of designing good classifiers. *SIAM Journal on Computing*, 30(1):318–323, 2000.

David P. Helmbold and Robert E. Schapire. Predicting nearly as well as the best pruning of a decision tree. *Machine Learning*, 27(1):51–68, 1997.

Matti Kääriäinen and Tapio Elomaa. Rademacher penalization over decision tree prunings. In Nada Lavrač, Dragan Gamberger, Hendrik Blockeel, and Ljupčo Todorovski, editors, *Machine Learning: ECML 2003, Proceedings of the Fourteenth European Conference*, volume 2837 of *Lecture Notes in Artificial Intelligence*, pages 193–204, Berlin Heidelberg New York, 2003. Springer.

Michael Kearns and Yishay Mansour. A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In Jude Shavlik, editor, *Proceedings of the Fifteenth International Conference on Machine Learning*, pages 269–277, San Francisco, CA, 1998. Morgan Kaufmann.

Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

Vladimir Koltchinskii and Dmitry Panchenko. Rademacher processes and bounding the risk of function learning. In Evarist Giné, David M. Mason, and Jon A. Wellner, editors, *High Dimensional Probability II*, pages 443–459. Birkhäuser, Boston, 2000.

John Langford. Combining training set and test set bounds. In Claude Sammut and Achim G. Hoffmann, editors, *Proceedings of the Nineteenth International Conference on Machine Learning*, pages 331–338, San Francisco, CA, 2002. Morgan Kaufmann.

John Langford. Practical prediction theory for classification, 2003. A tutorial presented at ICML 2003. Available at `http://hunch.net/˜jl/projects/prediction_bounds/tutorial/tutorial.pdf`.

John Langford and David McAllester. Computable shell decomposition bounds. In Nicolò Cesa-Bianchi and Sally A. Goldman, editors, *Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 25–34, San Francisco, CA, 2000. Morgan Kaufmann.

Fernando Lozano. Model selection using Rademacher penalization. In *Proceedings of the Second ICSC Symposium on Neural Networks*, Berlin, 2000. NAISO Academic Press.

Gábor Lugosi and Marten Wegkamp. Complexity regularization via localized random penalties. *The Annals of Statistics*, 32(4):1679–1697, 2004.

Yishay Mansour. Pessimistic decision tree pruning based on tree size. In Douglas H. Fisher, editor, *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 195–201, San Francisco, CA, 1997. Morgan Kaufmann.

Pascal Massart. Some applications of concentration inequalities to statistics. *Annales de la Faculté des Sciences de Toulouse*, IX:245–303, 2000.

Colin McDiarmid. On the method of bounded differences. In J. Siemons, editor, *Surveys in Combinatorics*, volume 141 of *London Mathematical Society Lecture Note Series*, pages 148–188. Cambridge University Press, 1989.

Shahar Mendelson and Petra Philips. Random subclass bounds. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines, Proceedings of the Sixteenth Annual Conference on Learning Theory and Seventh Kernel Workshop, COLT/Kernel 2003*, volume 2777 of *Lecture Notes in Artificial Intelligence*, pages 329–343, Berlin Heidelberg New York, 2003. Springer.

John Mingers. An empirical comparison of pruning methods for decision tree induction. *Machine Learning*, 4(2):227–243, 1989.

Jonathan J. Oliver and David J. Hand. On pruning and averaging decision trees. In Armand Prieditis and Stuart Russell, editors, *Proceedings of the Twelfth International Conference on Machine Learning*, pages 430–437, San Francisco, CA, 1995. Morgan Kaufmann.

Francesco C. Pereira and Yoram Singer. An efficient extension to mixture techniques for prediction and decision trees. *Machine Learning*, 36(3):183–199, 1999.

J. Ross Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27(3): 221–248, 1987.

J. Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1993.

Aad W. Van der Vaart and Jon A. Wellner. *Weak Convergence and Empirical Processes*. Springer, New York, 2000. Corrected second printing.

Vladimir N. Vapnik. *Estimation of Dependencies Based on Empirical Data*. Springer, New York, 1982.

Vladimir N. Vapnik and Alexey Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications*, 16(2):264–280, 1971.

Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, CA, 1999.