Rademacher Penalization over Decision Tree Prunings

Matti Kääriäinen and Tapio Elomaa

Department of Computer Science, University of Helsinki, Finland
{matti.kaariainen,elomaa}@cs.helsinki.fi

Abstract. Rademacher penalization is a modern technique for obtaining data-dependent bounds on the generalization error of classifiers. It would appear 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. Moreover, we generalize the error-bounding approach from binary classification to multi-class situations. Our empirical experiments indicate that the proposed new bounds clearly outperform earlier bounds for decision tree prunings and provide non-trivial error estimates on real-world data sets.

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 for all imaginable situations. Using *Rademacher complexity* [1,2] 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 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 [3]. We have applied Rademacher penalization to two-level decision trees [4], which can be learned efficiently in the agnostic PAC model [5].

General decision tree growing algorithms are necessarily heuristic because of the computational complexity of finding optimal decision trees [6]. 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 [7] produces results that are very competitive in prediction accuracy with better motivated approaches. We consider the usual two-phase process of decision tree learning;

N. Lavrač et al. (Eds.): ECML 2003, LNAI 2837, pp. 193–204, 2003.

after growing a tree, it is pruned in order to reduce its dependency on the training data and to better reflect characteristics of future data. By the practical success of decision tree learning, prunings of an induced decision tree have to 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 training data to produce a decision tree. Then a pruning algorithm, for example the *reduced error pruning* (Rep) algorithm of Quinlan [8], 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, 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.

Our empirical experiments show that Rademacher penalization applied to prunings found by Rep provides reasonable generalization error bounds on real-world data sets. Although the bounds still overestimate the test set error, they are much tighter than the earlier distribution independent bounds for prunings.

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 extend to cover the multi-class case. Section 3 concerns pruning of decision trees, reduced error pruning of decision trees being the main focus. Related pruning approaches are briefly reviewed 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 $h \in \mathcal{H}$ that may 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:

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

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:

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

where ℓ is the 0/1 loss function for which $\ell(y, y') = 1$ if $y \neq y'$ and 0 otherwise.

Empirical Risk Minimization (ERM) [9] 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 $\sup_{h \in \mathcal{H}} |\epsilon_P(h) - \hat{\epsilon}_n(h)|$. 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. 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 [1]. 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

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

The following symmetrization inequality [10], which covers also the multi-class setting, connects Rademacher penalties to generalization error analysis.

Theorem 1. *The inequality*

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

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

The random variables $\sup_{h \in \mathcal{H}} |\epsilon_P(h) - \hat{\epsilon}_n(h)|$ and $R_n(\mathcal{H})$ are sharply concentrated around their expectations [1]. The concentration results are based on the following McDiarmid's bounded difference inequality [11].

Lemma 1 (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 $\varepsilon > 0$

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

are upper bounded by $\exp(-2\varepsilon^2 / \sum_{i=1}^n c_i^2)$.

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

Lemma 2. *Let $h \in \mathcal{H}$ be arbitrary. Then with probability at least $1 - \delta$*

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

where $\eta(\delta, n) = \sqrt{\ln(2/\delta)/(2n)}$ is a small 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 $2/n$. Thus, Lemma 1 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

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

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

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

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

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

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

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

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

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

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 [1] 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:

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

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 $z = \bar{y}$ 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

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

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 $\bar{y}_i$ 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 $\bar{z}_i$, respectively. Here, we follow the convention that $\bar{\bar{z}} = z$ for all $z \in \mathcal{Y} \cup \overline{\mathcal{Y}}$.
- The Rademacher penalty is given by the maximum of $|\{ i : r_i = +1 \}|/n - \hat{\epsilon}(h_1)$ and $|\{ i : r_i = -1 \}|/n - \hat{\epsilon}(h_2)$, where the empirical errors $\hat{\epsilon}(h_1)$ and $\hat{\epsilon}(h_2)$ are with respect to the labels z_i and $\bar{z}_i$, respectively.

In the two-class setting, the set $\bar{y}$ of all classes but y, $\mathcal{Y} \setminus \{ y \}$, is a singleton. Thus, changing class y to $\bar{y}$ 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 Rep nor with T2, a decision tree learning algorithm used in our earlier study, since both empirical risk minimization algorithms can easily be adapted to handle this more general setting as explained in the next section for Rep and argued by Auer et al. [5] for T2.

3 Growing and Pruning Decision Trees

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. Many heuristic approaches [8,12,13] as well as more analytical ones [14,15] to pruning have been proposed. A special class of pruning algorithms are the on-line ones [16,17]. 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 [8]. It has been used rather rarely in practical learning algorithms mainly because it requires part of the available data to be reserved solely for pruning purposes. However, empirical evaluations of pruning algorithms indicate that the performance of Rep is comparable to other more widely used pruning strategies [12,13]. In analyses Rep has often been considered a representative pruning algorithm [13,18]. It produces an optimal pruning of a given tree — the smallest tree among those with minimal error (with respect to the set of pruning examples) [13,19].

Table 1 presents the Rep algorithm in pseudo code (for simplicity only for decision trees with binary splits). It works in two phases: First the set of pruning examples S is classified using the given tree T to be pruned. The node statistics 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.

The scarceness of (expensive) data used to be considered a major problem facing inductive algorithms. Therefore, Rep's requirement of a separate pruning set of examples has been seen prohibitive. Nowadays the situation has turned around: In data mining abundance of data is considered to be a major problem for learning algorithms to cope with. Thus, it should not be a major obstacle to leave some part of the data aside from the decision tree building phase and to reserve it for pruning purposes.

Rep is an ERM algorithm for the hypothesis class consisting of all prunings of a given decision tree (for a proof, see [19]). 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 (C4.5) to grow a decision tree based on a set of training examples. The tree serves as a representation of the data-dependent hypothesis class that consists of its prun-

Table 1. The REP algorithm capable of handling complement labels also. The algorithm first classifies the pruning examples in a top-down pass using method `classify` and then, during a bottom-up pass, prunes the tree using method `prune`

```
decTree REP( decTree T, exArray S ) // Prune the tree
  for( i= 0 to S.length-1 ) classify( T, S[i] );
  prune( T ); return T;

void classify( decTree T, example e ) // Update node counters top-down
  T.total++; T.count[e.label]++;
  if( !leaf(T) )
    if( T.test(e)==0 ) classify( T.left, e );
    else classify( T.right, e );

int error( label y, cntArray count ) // Compute classification error
  int errors= 0;
  foreach( z in Y-{y} ) errors+= count[z];
  return errors + count[bar(y)];

int prune( decTree T ) // Output classification error after pruning
  int leafError= error( T.label, T.count );
  if( leaf(T) ) return leafError;
  int treeError= prune( T.left )+ prune( T.right );
  if( treeError < leafError ) return treeError;
  else replace T with a leaf labeled T.label;
    return leafError;
```

ings. 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 as well. However, since REP is an empirical risk minimization algorithm, the derived error bounds will be the tightest when combined with it.

Our view on pruning is similar to that of Esposito et al. [20], who viewed many decision tree pruning algorithms as instantiations of search in the state space consisting of all prunings of a given decision tree, the state transition function being determined by the basic pruning operation. In this setting, REP can be seen as a search algorithm whose bias is determined by the ERM principle and the tendency to favor small hypotheses. Our goal, however, is not to analyze the search itself, but to evaluate the goodness of the final pruning produced by the search algorithm. We pursue this goal further in Section 5.

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 training and the rest for pruning has been observed to work well in practice [13]. If the initial data set is very large, it may be computationally infeasible to use all the data for training or pruning. In that case one can use heuristic sequential sampling methods for selecting the size of the training set and determine the size of the pruning set, e.g., by using progressive Rademacher sampling [4]. Because Rep is an efficient linear-time algorithm, it is not hit hard by overestimated pruning sample size.

4 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 [21,22,23,24]. However, often optimality is measured over the training set. Then it is only possible to maintain the initial training set accuracy, assuming that no noise is present. Neither is it usually possible to reduce the size of the decision tree without sacrificing the classification accuracy. For example, Bohanec and Bratko [22] as well as Almuallim [24] 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 [16,17]. 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 the 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 proven to be competitive with the best possible pruning of the initial tree. 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. 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 [14] and Kearns and Mansour [15] are very similar to Rep in spirit. The main difference with these pruning 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 [21] 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 [14] and Kearns and Mansour [15] provide generalization error analyses for their algorithms. The bound presented in [14] measures the complexity of the class of prunings by the size of the unpruned tree. 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. Mansour's bound can be used in connection with Rep, too, and we will use it as a point of comparison for our generalization error bounds in Section 6. Kearns and Mansour [15] 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.

5 Combining Rademacher Penalization and Decision Tree Pruning

When using 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 pruning found by Rep as if the hypothesis class from which Rep selects a pruning 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 (or any other pruning algorithm). Moreover, since Rep is a linear-time ERM algorithm for the class of prunings, it can be used to evaluate the generalization error bounds efficiently.

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

- Split the available data into a growing set and a pruning set.
- Use, e.g., C4.5 (without pruning) on the growing set to induce a decision tree.
- 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.
- Evaluate the error bound as explained in Section 2 by running Rep two more times.

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, 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 whenever C4.5 works well for the

data-generating distribution in question. The empirical evidence overwhelmingly demonstrates that C4.5 usually fares quite well.

Generalization error bounds can be roughly divided into two categories: Those based on a training set only and those requiring a separate test set [25]. Our generalization error bounds for prunings may be seen to lie somewhere between these two extremes. We use only part of the data in the tree growing phase. The rest — the set of pruning data — is used 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

The obvious performance reference for the approach of Rademacher penalization over decision tree prunings is to compare it to existing generalization error bounds such as the ones presented by Mansour [14] and Kearns and Mansour [15]. The bound in the latter is impossible to evaluate in practice because it requires knowing the depth and size of the pruning with the best generalization error. This leaves us with the bound of Mansour which only requires knowing the maximum size of prunings in advance. Bounds developed in the on-line pruning setting [16] are incomparable with the one presented in this paper because of the different learning model. Thus, they will not be considered here.

Mansour [14] derived, based on the Chernoff bound, the following bound for the generalization error of a decision tree h with k nodes:

$$\epsilon_P(h) < \hat{\epsilon}_n(h) + c\sqrt{\frac{k \log d + \log(2/\delta)}{n}},$$

where d is the arity of binary example vectors x_i and c is a constant. The bound applies only to binary decision trees in the two-class setting. When used for the class of unrestricted multi-class decision trees, the bound will give an overly optimistic estimate of what could be obtained with Mansour's proof technique in this more general setting. For the value of c we use a crude underestimate 0.5. Both these choices are in favor of Mansour's bound in the comparison.

The error bound based on Rademacher penalization depends on the data distribution so that its true performance can be evaluated only empirically. As benchmark data sets we use six large data sets from the UCI Machine Learning Repository, namely the Census income (2 classes), Connect (3 classes), Covertype (7 classes), and generated LED datasets (10 classes) with 5, 10, and 15 percent attribute noise and 300,000 instances. 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. [13].

Table 2. Averages and standard deviations of sizes of trees grown by C4.5 (left) and error bounds for REP (right) over 10 random splits of the data sets

Data set	Unpruned	Default	REP	Test set	R-bound	M-bound
Census	19732 ±732	1377 ±268	4749 ±397	4.9 ±0.1	8.7 ±0.2	49.9 ±0.9
Connect	10973 ±361	4253 ±104	4338 ±235	20.7 ±0.8	32.4 ±0.4	89.3 ±1.5
Cover	25356 ±221	22095 ±228	17404 ±179	6.9 ±0.1	12.7 ±0.1	44.0 ±0.2
LED24-5	27357 ±139	7042 ±74	3850 ±233	13.4 ±0.2	19.7 ±0.2	61.3 ±0.2
LED24-10	51790 ±204	13624 ±220	7671 ±323	26.4 ±0.1	36.8 ±0.2	91.7 ±0.2
LED24-15	71162 ±156	20273 ±259	11344 ±265	38.6 ±0.2	52.2 ±0.2	114.6 ±0.2

Table 2 summarizes the results of our experiments averaged over ten random splits of the data sets. 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 the default pruning of C4.5 and REP both manage to decrease the tree sizes considerably.

The right-hand side of Table 2 presents the test set accuracies and error bounds for REP prunings based on Rademacher penalization and Mansour's method. In both bounds, we set $\delta = 0.01$. Even though the bounds based on Rademacher 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. The Rademacher bounds are clearly superior to even the underestimates of the bounds by Mansour that we used as a benchmark. The amount by which the Rademacher bound overestimates the test set error is seen to be almost an order of magnitude smaller than the corresponding quantity related to Mansour's bound.

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 significantly tighter than previous generalization error bounds for decision tree prunings. However, the new bounds still appear unable to faithfully describe the performance attained in practice.

References

1. Koltchinskii, V.: Rademacher penalties and structural risk minimization. IEEE Trans. Inf. Theor. **47** (2001) 1902–1914
2. Bartlett, P.L., Mendelson, S.: Rademacher and Gaussian complexities: Risk bounds and structural results. JMLR **3** (2002) 463–482

3. Lozano, F.: Model selection using Rademacher penalization. In: Proc. 2nd ICSC Symposium on Neural Networks, NAISO Academic Press (2000)
4. Elomaa, T., Kääriäinen, M.: Progressive Rademacher sampling. In: Proc. 18th National Conference on Artificial Intelligence, MIT Press (2002) 140–145
5. Auer, P., Holte, R.C., Maass, W.: Theory and application of agnostic PAC-learning with small decision trees. In: Proc. 12th International Conference on Machine Learning, Morgan Kaufmann (1995) 21–29
6. Grigni, M., Mirelli, V., Papadimitriou, C.H.: On the difficulty of designing good classifiers. SIAM J. Comput. **30** (2000) 318–323
7. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
8. Quinlan, J.R.: Simplifying decision trees. Int. J. Man-Mach. Stud. **27** (1987) 221–248
9. Vapnik, V.N.: Estimation of Dependencies Based on Empirical Data. Springer (1982)
10. Van der Vaart, A.W., Wellner, J.A.: Weak Convergence and Empirical Processes. Springer (2000) Corrected second printing.
11. McDiarmid, C.: On the method of bounded differences. In: Surveys in Combinatorics. Volume 141 of London Mathematical Society Lecture Note Series. Cambridge University Press (1989) 148–188
12. Mingers, J.: An empirical comparison of pruning methods for decision tree induction. Mach. Learn. **4** (1989) 227–243
13. Esposito, F., Malerba, D., Semeraro, G.: A comparative analysis of methods for pruning decision trees. IEEE Trans. Pattern Anal. Mach. Intell. **19** (1997) 476–491
14. Mansour, Y.: Pessimistic decision tree pruning based on tree size. In: Proc. 14th International Conference on Machine Learning, Morgan Kaufmann (1997) 195–201
15. Kearns, M., Mansour, Y.: A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In: Proc. 15th International Conference on Machine Learning, Morgan Kaufmann (1998) 269–277
16. Helmbold, D.P., Schapire, R.E.: Predicting nearly as well as the best pruning of a decision tree. Mach. Learn. **27** (1997) 51–68
17. Pereira, F.C., Singer, Y.: An efficient extension to mixture techniques for prediction and decision trees. Mach. Learn. **36** (1999) 183–199
18. Oates, T., Jensen, D.: Toward a theoretical understanding of why and when decision tree pruning algorithms fail. In: Proc. 16th National Conference on Artificial Intelligence, MIT Press (1999) 372–378
19. Elomaa, T., Kääriäinen, M.: An analysis of reduced error pruning. J. Artif. Intell. Res. **15** (2001) 163–187
20. Esposito, F., Malerba, D., Semeraro, G.: Decision tree pruning as a search in the state space. In: Proc. 6th European Conference on Machine Learning. Volume 667 of Lecture Notes in Artificial Intelligence., Springer (1993) 165–184
21. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth (1984)
22. Bohanec, M., Bratko, I.: Trading accuracy for simplicity in decision trees. Mach. Learn. **15** (1994) 223–250
23. Oliver, J.J., Hand, D.J.: On pruning and averaging decision trees. In: Proc. 12th International Conference on Machine Learning, Morgan Kaufmann (1995) 430–437
24. Almuallim, H.: An efficient algorithm for optimal pruning of decision trees. Artif. Intell. **83** (1996) 347–362
25. Langford, J.: Combining training set and test set bounds. In: Proc. 19th International Conference on Machine Learning, Morgan Kaufmann (2002) 331–338