K-NORM MISCLASSIFICATION RATE ESTIMATION FOR DECISION TREES

Mingyu Zhong
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL, USA
email: myzhong@ucf.edu

Michael Georgiopoulos
School of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL, USA
email: michaelg@mail.ucf.edu

Georgios C. Anagnostopoulos
Department of Electrical and Computer Engineering
Florida Institute of Technology
Melbourne, FL, USA
email: georgio@fit.edu

ABSTRACT
The decision tree classifier is a well-known methodology for classification. It is widely accepted that a fully grown tree is usually over-fit to the training data and thus should be pruned back. In this paper, we analyze the overtraining issue theoretically using an the k-norm risk estimation approach with Lidstone's Estimate. Our analysis allows the deeper understanding of decision tree classifiers, especially on how to estimate their misclassification rates using our equations. We propose a simple pruning algorithm based on our analysis and prove its superior properties, including its independence from validation and its efficiency.

KEY WORDS
decision tree, pruning, Law of Succession

1 Introduction

Prediction and decision making are frequently used in real world applications. In making decisions, sometimes we need to know not only the best decision (the one with the least risk) but also the associated risk or, if the risk is not deterministic, the range of the risk. For example, compare the following outputs regarding the class of an input that is presented to the classifier.

1. Choosing class 1 has the lowest misclassification rate;
2. Choosing class 1 has the lowest misclassification rate that is approximately 0.4;
3. Choosing class 1 has the lowest misclassification rate that lies in the range 0.6 ± 0.1.

Obviously, the last classifier gives the most complete information among the three. In this paper, we focus on decision tree classifiers. Through the analysis conducted in this paper we are not only able to provide the optimal tree prediction, but also able to calculate the reliability of this prediction. Furthermore, we propose a pruning algorithm, referred to as k-norm pruning algorithm, which has the following properties: it has clear theoretical interpretation, it does not require cross-validation or a separate validation set, it can find the optimal tree within one traversal of the tree, and it has a simple implementation.

The rest of this paper is organized as follows. We first provide a literature review on decision tree error rate estimation and pruning (Section 2). Then, the necessary background knowledge on Lidstone's Law of Succession is covered in Section 3. In Section 4, we apply Lidstone's Law of Succession in the estimation of misclassification rates in decision trees, as well as the corresponding variance that serves as an indicator of the reliability of the output. Based on our analysis, we introduce the appropriate equations to estimate the tree's prediction accuracy and a pruning algorithm with the properties, mentioned above. We summarize our work in Section 5.

2 Related Work

One of the most classical decision trees is CART, proposed by Breiman et al., which established the framework (including the impurity measures and the pruning process) for future decision trees [1]. C4.5, one of the successors of CART, was introduced nine years later and has been widely used as a decision tree classifier to solve a variety of problems [16]. Although many issues have been raised since the introduction of decision trees, in this paper we only focus on the pruning and the related error rate estimations.

It is well-known that a classifier over-adapted to the training set tends to generalize poorly, when it is confronted with unseen instances. For a tree, although one could grow a full tree to obtain nearly 100% accuracy on the training data set, the tree would contain so many redundant nodes (those covering only a few instances) that the decision boundaries are far from optimal. On the other hand, Breiman et al., pointed out that it is difficult to set a stopping criterion to terminate the growing process to avoid generating redundant nodes (see [1], page 37). The main reason is the difficulty to foresee the performance of fu-

ture splits compared to current ones, which implies that no proper thresholds can be used to decide whether to stop. Thus, it has been widely accepted that one should grow the tree to the full size and prune it back. A number of pruning algorithms have been proposed, such as the Minimal Cost-Complexity Pruning (CCP) in CART (see [1], page 66), the Minimum Error Pruning (MEP) (see [15, 2]), the Error Based Pruning (EBP) in C4.5 (see [16], page 37), the Reduced Error Pruning (REP), the Pessimistic Error Pruning (PEP) (see [17] for both REP and PEP), the MDL-Based Pruning [14], the Classifiability Based Pruning [3], the Pruning using Backpropagation Neural Networks [9], etc. Some of the pruning algorithms are briefly analyzed and empirically compared in [4].

The ultimate problem in tree pruning is the estimation of the accuracy of unseen data. An independent validation set or cross-validation is one way to remedy this problem (see CCP [1] and REP [17] for example). Nevertheless, when the database is small, k-fold cross validation is not reliable either; when the database is large, this approach is computationally complex. For this reason, various estimations of error rates without validation are proposed (such as EBP [16], and MEP [15]) and compared (e.g., see [4]).

In the last decade, most researchers focused on the maximum likelihood estimation with confidence intervals (see [16, 12, 5, 8, 13, 7]); they produced a considerable number of results in computing the confidence upper bound of the error rate and the application in tree pruning. Most of their results do not make any assumption on the data distribution and do not require any validation, due to the nature of the maximum likelihood estimation. In their approaches, however, the confidence intervals are usually misinterpreted: if [0.8,0.9] is a 90% confidence interval of the error rate, it does not mean that at 90% probability the error rate is between 0.8 and 0.9 (see [18], page 342 for a more detailed clarification). Therefore, any confidence interval is merely a heuristic to estimate the error rate.

Some other researchers applied a-posterior estimations rather than the maximum likelihood estimation. A typical example is MEP [15, 2], where the authors applied Lidstone's Law of Succession, which is a widely accepted alternative to the maximum likelihood estimation in statistics and is derived from the Bayesian theories assumption Dirichlet prior distributions. The authors use the a-posterior expected value to estimate the error rate. Unfortunately it turns out that MEP tends to under-prune a tree (see [4]). We argue that the application of Lidstone's Law of Succession is a reasonable choice since it yields interpretable results rather than heuristics, but the a-posterior expected value alone is too optimistic because it does not represent any reliability (see Subsection 4.4 for an example and Theorem 2 for a mathematical statement). Based on this observation, we propose the k-norm estimation to incorporate reliability metrics, with both theoretical interpretation and useful properties (e.g., no validation is required, and the optimal pruned tree can be found with a single traversal of the tree).

3 Lidstone's Law of Succession

An important issue in decision trees, as well as many other classifiers, is the estimation of the class probabilities given a set of training instances. To be more general, Let $\Phi_h (h = 1, 2, \cdots, H)$ be H *mutually exclusive and collectively exhaustive* events. Let p_h denote $P[\Phi_h]$ and N_h denote the number of occurrences of Φ_h in N independent trials. Obviously, N_h is a random variable. Suppose in N independent trials we observed n_h occurrences of Φ_h (an example would be observing n_j out of N training instances that belong to class j).

Lidstone's Law of Succession represents a-posteriori estimation, in which $p_h = P[\Phi_h]$ is estimated by p_h^*, where

$$p_h^* = E\left[p_h | n_1, \cdots, n_H\right] = \frac{n_h + \lambda}{N + \lambda H}, \tag{1}$$

where λ is a predefined non-negative parameter. In statistics, a widely used value for λ is 0.5 (Krichevskiy suggests the value 0.50922 in [11]).

Lidstone's Estimation is widely used in probability estimation, such as in Naive Bayes Classifiers [10]. However, most researchers simply compute the expected value according to (1) while ignoring the variance. In Theorem 2, we will show that the expected value (1-norm) is not sufficient to correctly evaluate a tree's accuracy on unseen data.

In Lidstone's Estimation, the a-prior distribution of p_h is assumed under Dirichlet distribution, denoted by $(p_1, \cdots, p_H) \sim Dir(\lambda, \lambda, \cdots, \lambda)$, or

$$f\left(p_1, \cdots, p_H\right) = \alpha\delta\left(1 - \sum_{h=1}^{H} p_h\right) \prod_{h=1}^{H} p_h^{\lambda-1} I[p_h \geq 0], \tag{2}$$

where α is a constant so that the integral of $f\left(p_1, \cdots, p_H\right)$ is unity, and $\delta()$ is the Dirac delta function.

Based on (2), it has been proven that the a-posterior probabilities also follow the Dirichlet distribution in [6]. In particular, $(p_1, \cdots, p_H | n_1, \cdots, n_H) \sim Dir(n_1+\lambda, \cdots, n_H+\lambda)$, and $(p_h, 1 - p_h | n_1, \cdots, n_H) \sim Dir(n_h + \lambda, N - n_h + \lambda(H - 1))$. This result leads to

$$E\left[p_h^k \middle| n_1, \cdots, n_H\right] = \prod_{m=0}^{k-1} \frac{n_h + \lambda + m}{N + \lambda H + m}, \tag{3}$$

and

$$\begin{aligned} & E\left[(1 - p_h)^k \middle| n_1, \cdots, n_H\right] \\ = & \prod_{m=0}^{k-1} \frac{N - n_h + \lambda(H - 1) + m}{N + \lambda H + m}. \end{aligned} \tag{4}$$

The variance of p_h can be computed as

$$\begin{aligned} & Var\left[p_h | n_1, \cdots, n_H\right] \\ = & E\left[p_h^2 \middle| n_1, \cdots, n_H\right] - E\left[p_h | n_1, \cdots, n_H\right]^2. \end{aligned} \tag{5}$$

Equation (4) is the basis of our proposed error rate estimation and the corresponding k-norm pruning algorithm.

4 Error Rate Estimation in Decision Trees

If we apply the Maximum Likelihood Estimation to evaluate a decision tree, we cannot solve the overtraining problem because the estimated misclassification rate would be zero if the tree is grown to the full size (each leaf having a pure class label). Therefore, we apply the Lidstone's Law of Succession.

4.1 Terminologies

4.1.1 Events

In a tree, let A_t denote the event that the node t is activated (it receives the input). Apparently, if c is a child of t, A_c implies A_t (that is, $(A_c, A_t) = A_c$).

Furthermore, let J be the number of classes and C_j ($j = 1, 2, \cdots, J$) denote the event that the attribute vector $\mathbf{X}$ is associated with class j.

4.1.2 Error rates

Let Err be the event of misclassification. Apparently this event depends on the input attribute vector $\mathbf{X}$. We now estimate the misclassification rate $P[Err|\mathbf{X}]$ of a tree, by treating $P[Err|\mathbf{X}]$ as a random variable.

Since the root of the tree is always activated,

$$P[Err|\mathbf{X}] = P[Err|A_{root}, \mathbf{X}]. \tag{6}$$

In general, we define the error rate of a node t as

$$r_t = P[Err|A_t, \mathbf{X}]. \tag{7}$$

For a leaf, apparently

$$r_t = 1 - P[C_{Label(t)}|A_t, \mathbf{X}], \tag{8}$$

$$Label(t) = \arg\max_j n_{j,t}, \tag{9}$$

where $n_{j,t}$ is the number of training examples of class j in node t. Note that $Label(t)$ is a constant number once the tree is grown.

4.1.3 Expected Values

In the rest of this section we use expected values extensively, and thus it worthy mentioning first that the expected value here is the integral across two random factors: the random input attribute vector $\mathbf{X}$ and the unknown class probabilities $\mathbf{P}$. That is, for any random variable Q,

$$E_{\mathbf{X}}[Q] = \int Qf(\mathbf{X})d\mathbf{X}, \tag{10}$$

$$E_{\mathbf{P}}[Q] = \int Qf(\mathbf{P})d\mathbf{P}, \tag{11}$$

$$E[Q] = E_{\mathbf{X}}[E_{\mathbf{P}}[Q]] = E_{\mathbf{P}}[E_{\mathbf{X}}[Q]]. \tag{12}$$

Similarly,

$$VAR_{\mathbf{P}}[Q] = E_{\mathbf{P}}\left[Q^2\right] - E_{\mathbf{P}}[Q]^2, \tag{13}$$

$$VAR[Q] = E\left[Q^2\right] - E[Q]^2. \tag{14}$$

When conditions are involved, we put the conditions into the subscript. For example,

$$E_{\mathbf{X}|A_t}[Q] = \int Qf(\mathbf{X}|A_t)d\mathbf{X}. \tag{15}$$

Note that $\mathbf{P}$ is composed of all probabilities, including $P[C_j|A_t]$ for any node t, and the expression "$P[C_j|A_t]|A_t$" is not meaningful. Therefore, we treat $\mathbf{P}$ as independent of A_t and compute the expected value of a random variable within a node t as

$$E^t[Q] = E_{\mathbf{P}}\left[E_{\mathbf{X}|A_t}[Q]\right], \tag{16}$$

where the superscript t represents the condition "$|A_t$". When Q has the same subscript t, we omit the superscript t in $E^t[Q]$. For example, the expected error rate of the subtree rooted at t is represented by $E[r_t]$ rather than $E^t[r_t]$, because this expected value must, of course, be computed with the assumption that A_t is true.

Note that in the rest of this paper, **all expected values are conditional (given the observations with the training examples**. We omit the conditions only for simplicity in our equations. Please keep in mind that "$E[r_t^k]$" actually refers to "$E[r_t^k|Observations]$".

4.2 Assumptions

We make the following assumptions in our analysis:

1. We assume that the children of the same node t are mutually exclusive, which means that for this decision node,

$$\begin{aligned} & P[Err|A_t, \mathbf{X}] \\ = & \sum_{c \in Children(t)} P[Err|A_c, A_t, \mathbf{X}]P[A_c|A_t, \mathbf{X}] \\ = & \sum_{c \in Children(t)} P[Err|A_c, \mathbf{X}]P[A_c|A_t, \mathbf{X}]. \end{aligned} \tag{17}$$

   The above equations indicate that the error rate of a decision tree can be evaluated recursively.

2. We assume no missing data in $\mathbf{X}$; given an input $\mathbf{X}$, only one deterministic child is activated. That is, $P[A_c|A_t, \mathbf{X}]$ is either zero or one. This assumption implies that

$$P[A_c|A_t, \mathbf{X}]^k = P[A_c|A_t, \mathbf{X}], \quad \forall k > 0, \tag{18}$$

$$P[A_{c_1}|A_t, \mathbf{X}]P[A_{c_2}|A_t, \mathbf{X}] = 0, \quad \text{if } c_1 \neq c_2. \tag{19}$$

Now we rely on a theorem (Section 4.3) and an example (Section 4.4) to introduce the k-norm estimation approach. The theorem provides an efficient way to compute the expected values and the variance of the risks.

4.3 Partitioning Theorem

Theorem 1 *Under the assumptions in Subsection 4.2, for any decision node d and any natural number k,*

$$E_{\mathbf{X}|A_d}\left[r_d^k\right] = \sum_{c \in Children(d)} E_{\mathbf{X}|A_c}\left[r_c^k\right] P\left[A_c|A_d\right], \tag{20}$$

where r_t is the error rate of node t, defined as follows:

$$r_t = P\left[Err|A_t, \mathbf{X}\right]. \tag{21}$$

The proof of this theorem is fairly simple using (18) and (19). Due to the limited space, all proofs in this paper are omitted.

Corollary 1 *Under the assumptions in Subsection 4.2, as well as the assumption that r_c and $P\left[A_c|A_d\right]$ are independent random variables for each child node c of the decision node d, the following expression is valid:*

$$E\left[r_d^k\right] = \sum_{c \in Children(d)} E\left[r_c^k\right] E_{\mathbf{P}}\left[P\left[A_c|A_d\right]\right]. \tag{22}$$

This corollary can be proven by computing the expected values with respect to $\mathbf{P}$ of both sides of (20), according to (16). Since $\mathbf{P}$ includes $P[A_c|A_d]$, the other probabilities in $\mathbf{P}$ do not contribute to $E_{\mathbf{P}}\left[P\left[A_c|A_d\right]\right]$ (because $E_Y[E_Z[Z]] = E_Z[Z]$ for any two random variables Y and Z even if they are dependent on each other). Recall that all expected values here are conditional, based on the training examples. It is well known that to estimate the probabilities of a set of mutually exclusive and collectively exhaustive events with N independent trials, the numbers of occurrences of the events are sufficient. Therefore, the condition "given observations" can be replaced with "given n_c for all $c \in Children(d)$" (where n_t is the number of training examples covered by node t). Using Lidstone's Law of Succession, we get

$$E_{\mathbf{P}}\left[P\left[A_c|A_d\right]\right] = \frac{n_c + \eta}{n_d + \eta K_d}, \tag{23}$$

where K_d is the number of immediate children of decision node d, and η has an analogous functionality as λ (we use a different notation because λ will be used to denote the parameter in the error rate estimation). For simplicity, from now on, we use $P^*\left[A_c|A_d\right]$ to denote $E_{\mathbf{P}}\left[P\left[A_c|A_d\right]\right]$. Using the corollary, the variance of the error rate can be computed as follows.

$$E\left[r_d\right] = \sum_{c \in Children(d)} E\left[r_c\right] P^*\left[A_c|A_d\right], \tag{24}$$

$$E\left[r_d^2\right] = \sum_{c \in Children(d)} E\left[r_c^2\right] P^*\left[A_c|A_d\right], \tag{25}$$

$$VAR\left[r_d\right] = E\left[r_d^2\right] - E\left[r_d\right]^2. \tag{26}$$

It is important to note that generally,

$$VAR\left[r_d\right] \neq \sum_{c \in Children(d)} VAR\left[r_c\right] P^*\left[A_c|A_d\right]. \tag{27}$$

4.4 An Example

Consider now a more specific example of a 2-class classification problem for which a binary tree is built. We focus on the error rate, and thus the risk r_t is defined as $P\left[Err|A_t, \mathbf{X}\right]$. Suppose a node t of the tree receives 98 training examples of class 1 and 1 example of class 2, and a split separates the two classes. Assume $\lambda = \eta = 0.5$. According to (4) and (8), before splitting,

$$E\left[r_t\right] \approx E_{\mathbf{P}}\left[r_t\right] = \frac{1 + 0.5}{99 + 0.5 \times 2} = 0.015000, \tag{28}$$

$$\begin{aligned} E\left[r_t^2\right] &\approx E_{\mathbf{P}}\left[r_t^2\right] \\ &= \frac{(1 + 0.5)(1 + 0.5 + 1)}{(99 + 0.5 \times 2)(99 + 0.5 \times 2 + 1)} \\ &= 0.00037129, \end{aligned} \tag{29}$$

$$VAR\left[r_t\right] \approx E_{\mathbf{P}}\left[r_t^2\right] - E_{\mathbf{P}}\left[r_t\right]^2 = (0.012095)^2. \tag{30}$$

After splitting, for the left child,

$$E\left[r_{t_1}\right] \approx \frac{0 + 0.5}{98 + 0.5 \times 2} = 0.0050505, \tag{31}$$

$$\begin{aligned} E\left[r_{t_1}^2\right] &\approx \frac{(0 + 0.5)(0 + 0.5 + 1)}{(98 + 0.5 \times 2)(98 + 0.5 \times 2 + 1)} \\ &= 0.00075758, \end{aligned} \tag{32}$$

$$P^*\left[A_{t_1}|A_t\right] = \frac{98 + 0.5}{99 + 0.5 \times 2} = 0.98500. \tag{33}$$

After splitting, for the right child,

$$E\left[r_{t_2}\right] \approx \frac{0 + 0.5}{1 + 0.5 \times 2} = 0.25000, \tag{34}$$

$$\begin{aligned} E\left[r_{t_2}^2\right] &\approx \frac{(0 + 0.5)(0 + 0.5 + 1)}{(1 + 0.5 \times 2)(1 + 0.5 \times 2 + 1)} \\ &= 0.12500, \end{aligned} \tag{35}$$

$$P^*\left[A_{t_2}|A_t\right] = \frac{1 + 0.5}{99 + 0.5 \times 2} = 0.015000. \tag{36}$$

For the sub-tree, according to Theorem 1,

$$\begin{aligned} E\left[r_t\right] &\approx 0.0050505 \times 0.985 + 0.25 \times 0.015 \\ &= 0.0087247, \end{aligned} \tag{37}$$

$$\begin{aligned} E\left[r_t^2\right] &\approx 0.00075758 \times 0.985 + 0.125 \times 0.015 \\ &= 0.0019496, \end{aligned} \tag{38}$$

$$\begin{aligned} VAR\left[r_t\right] &\approx 0.0019496 - (0.0087247)^2 \\ &= (0.04328)^2. \end{aligned} \tag{39}$$

Although the expected value of the error rate decreases by 0.006275 after the split, the average standard deviation increases by 0.03119, which is almost 5 times larger than the decreased amount in the expected value. Therefore, we do not expect that this split will improve the generalization.

4.5 k-Norm Estimation

The above example indicates that the expected value of the error rate alone is not a good estimate. In general, we have the following theorem:

Theorem 2 *If $0 \leq \eta \leq \lambda J$, $\lambda < 1/(K_t - 1)(J-1)$, and a split at a leaf reduces the training misclassifications, then the split decreases the expected error rate.*

We should consider the variance for a more realistic estimation of the risk. Although we could define the estimated risk r^* as $E[r] + \sqrt{VAR[r]}$ analogously to the 1-SE rule in CART (see [1], page 78), we observe in the above example that $E\left[r^2\right]$ is also a good indicator of the generalization. We prefer to use the k-norm estimation $\|r\|_k = \sqrt[k]{E[|r|^k]}$, because:

1. According to Theorem 1, $E\left[r^k\right]$ can be computed incrementally, by starting at the leaves of the tree and moving upwards towards the root of the tree. To optimize any sub-tree rooted at t, we can first optimize the sub-trees under t and then optimize the node t, which guarantees global optimality. This property is very important for our k-norm pruning algorithm;

2. When $k = 2$, $\|r\|_2 = \sqrt{E[r]^2 + VAR[r]}$ takes both the expected value and the variance into account.

As mentioned above, we recommend $k = 2$ for the k-norm estimation. We prefer smaller values of k (except for $k = 1$ which represents only the expected value) for the following reasons:

- Given a finite set of examples, when k is higher, the estimation of the k-norm becomes more sensitive to noise and computational errors, which can also be seen in the approximation in (45);

- According to (3) and (4), the computational complexity of the k-th moments is at least $O(k)$.

4.6 Application in Tree Prediction

Given an input $\mathbf{X}$, we cannot predict only the class label but we can also provide an estimated error rate for this prediction. Here $\mathbf{X}$ is deterministic and therefore we only compute $E_{\mathbf{P}}\left[P[Err|\mathbf{X}]^k\right]$ and the result depends on $\mathbf{X}$.

$$r = P[Err|\mathbf{X}] = 1 - P\left[C_{Label(t)}|\mathbf{X}, A_t\right], \qquad (40)$$

where t is the leaf where $\mathbf{X}$ falls. The above equation is intuitive and thus its derivation is omitted. Under the assumptions in Subsection 4.2,

$$\begin{aligned} E_{\mathbf{P}}\left[r^k\right] &= E_{\mathbf{P}}\left[\left(1 - P\left[C_{Label(t)}|\mathbf{X}, A_t\right]\right)^k\right] \\ &\approx E_{\mathbf{P}}\left[\left(1 - P\left[C_{Label(t)}|A_t\right]\right)^k\right]. \end{aligned} \qquad (41)$$

According to (4),

$$E_{\mathbf{P}}\left[r^k\right] \approx \prod_{i=0}^{k-1} \frac{n_t - \max_j n_{j,t} + \lambda(J-1) + i}{n_t + \lambda J + i}. \qquad (42)$$

In practice, there are two reasonable ways to output the estimated error rate: 1) output the realistic error rate $\|r\|_2 = \sqrt{E_{\mathbf{P}}[r^2]}$ only, and 2) output $E_{\mathbf{P}}[r]$ and $\sqrt{VAR_{\mathbf{P}}[r]} = \sqrt{E_{\mathbf{P}}[r^2] - E_{\mathbf{P}}[r]^2}$, the latter representing the reliability.

4.7 Application in Tree Pruning

Let

$$r_t = P[Err|A_t, \mathbf{X}]. \qquad (43)$$

Apparently, the error rate of the tree $r = r_{root}$. Under the assumptions in Subsection 4.2, for a decision node t,

$$E\left[r_t^k\right] = \sum_{c \in Children(t)} E\left[r_c^k\right] P^*[A_c|A_t]. \qquad (44)$$

For a leaf t,

$$\begin{aligned} E\left[r_t^k\right] &\approx E_{\mathbf{P}}\left[P[Err|A_t]\right] \\ &= \prod_{i=0}^{k-1} \frac{n_t - \max_j n_{j,t} + \lambda(J-1) + i}{n_t + \lambda J + i}. \end{aligned} \qquad (45)$$

By using the k-norm error rate estimation, in order to get the optimally pruned tree at t, we can first get the optimally pruned trees below t, and then consider whether pruning t yields a lower k-norm error rate. The algorithm is shown below:

input : a tree rooted at node t
output: the k-th moment error rate of the optimal pruned tree (modified from the input)

Let $R_{leaf} = \prod_{m=0}^{k-1} \frac{N_t - n_{Label(t),t} + \lambda(J-1) + m}{N_t + \lambda J + m}$;
if *t is a decision node* **then**
  Let $R_{tree} = \sum_{c \in Children(t)} \frac{N_c + \eta}{N_t + \eta K_t} \texttt{Prune}(c)$;
  if $\sqrt[k]{R_{tree}} < \sqrt[k]{R_{leaf}} - \varepsilon$ **then**
    return R_{tree};
  end
  Replace the subtree rooted at t with a leaf;
end
return R_{leaf};

Algorithm 1: PruneTree

We compared our pruning algorithm to Cost-Complexity Pruning of CART (see [1]) and Error-based Pruning of C4.5 (see [16]). The results, which are published in [19], show that our algorithm is superior in both accuracy (especially when the size of the training set is small) and speed (especially when the size of the training set is large).

5 Conclusions

In this paper, we carried out a theoretical analysis of the misclassification rate of decision tree classifiers. We showed that although a split tends to decrease the expected value of the misclassification rate, it might increase the variance of the misclassification rate (that is, weakens the reliability of the tree). We proposed a k-norm estimation algorithm that takes into account both the expected value and the variance and the resulting algorithms for estimating the prediction error rate and for pruning the tree. We also showed important properties of our k-norm pruning algorithm (e.g., it finds the optimally pruned tree in one traversal of the tree) by providing appropriate theorems and illustrative examples. Finally, it is important to note that our approach has far more general applications: Theorem 1 can also be applied to other measures of interests in decision tree classification (e.g., the risks can be redefined using appropriate misclassification costs).

Acknowledgment

This work was supported in part by the NSF grants: CRCD: 0203446, CCLI: 0341601, DUE: 05254209, IIS-REU: 0647120, and IIS-REU: 0647018.

References

[1] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Wadsworth, 1984.

[2] B. Cestnik and I. Bratko. On estimating probabilities in tree pruning. In *EWSL-91: Proceedings of the European working session on learning on Machine learning*, pages 138–150, New York, NY, USA, 1991. Springer-Verlag New York, Inc.

[3] M. Dong and R. Kothari. Classifiability based pruning of decision trees. In *IJCNN*, volume 3, pages 1739–1743, 2001.

[4] F. Esposito, D. Malerba, and G. Semeraro. A comparative analysis of methods for pruning decision trees. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(5):476–491, 1997.

[5] Y. Freund. Self bounding learning algorithms. In *COLT: Proceedings of the Workshop on Computational Learning Theory*. Morgan Kaufmann Publishers Inc., 1998.

[6] I. J. Good. *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. Number 30 in Research monographs. MIT Press, Cambridge, MA, 1965.

[7] M. Kääriäinen and T. Elomaa. Rademacher penalization over decision tree prunings. In N. Lavrac, D. Gamberger, L. Todorovski, and H. Blockeel, editors, *ECML*, volume 2837 of *Lecture Notes in Computer Science*, pages 193–204. Springer, 2003.

[8] M. J. Kearns and Y. Mansour. A fast, bottom-up decision tree pruning algorithm with near-optimal generalization. In *ICML '98: Proceedings of the Fifteenth International Conference on Machine Learning*, pages 269–277, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[9] B. Kijsirikul and K. Chongkasemwongse. Decision tree pruning using backpropagation neural networks. In *IJCNN*, volume 3, pages 1876–1880, 2001.

[10] R. Kohavi, B. Becker, and D. Sommerfield. Improving Simple Bayes. In M. van Someren and G. Widmer, editors, *ECML*, volume 1224 of *Lecture Notes in Computer Science*, pages 78–87. Springer, 1997.

[11] R. E. Krichevskiy. Laplace's law of succession and universal encoding. *IEEE Transactions on Information Theory*, 44(1):296–303, 1998.

[12] Y. Mansour. Pessimistic decision tree pruning based on tree size. In *Proc. 14th International Conference on Machine Learning*, pages 195–201. Morgan Kaufmann, 1997.

[13] Y. Mansour and D. A. McAllester. Generalization bounds for decision trees. In *COLT '00: Proceedings of the Thirteenth Annual Conference on Computational Learning Theory*, pages 69–74, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[14] M. Mehta, J. Rissanen, and R. Agrawal. MDL-based decision tree pruning. In *KDD*, pages 216–221, 1995.

[15] T. Niblett and I. Bratko. Learning decision rules in noisy domains. In *Proceedings of Expert Systems '86, the 6th Annual Technical Conference on Research and development in expert systems III*, pages 25–34, 1986.

[16] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, 1993.

[17] J. R. Quinlan. Simplifying decision trees. *Int. J. Hum.-Comput. Stud.*, 51(2):497–510, 1999.

[18] S. B. Vardeman and J. M. Jobe. *Basic Engineering Data Collection and Analysis.* Brooks/Cole Thompson Learning, 2001.

[19] M. Zhong, M. Georgiopoulos, and G. C. Anagnostopoulos. Experiments with an innovative tree pruning algorithm. In *IASTED International Conference on Artificial Intelligence and Applications*, pages 353–358, Innsbruck, Austria, 2007.