Machine Learning, 52, 199–215, 2003

Tree Induction for Probability-Based Ranking

FOSTER PROVOST provost@acm.org
New York University, New York, NY, USA

PEDRO DOMINGOS pedrod@cs.washington.edu
University of Washington, Seattle, WA, USA

Editor: Douglas Fisher

Abstract. Tree induction is one of the most effective and widely used methods for building classification models. However, many applications require cases to be ranked by the probability of class membership. Probability estimation trees (PETs) have the same attractive features as classification trees (e.g., comprehensibility, accuracy and efficiency in high dimensions and on large data sets). Unfortunately, decision trees have been found to provide poor probability estimates. Several techniques have been proposed to build more accurate PETs, but, to our knowledge, there has not been a systematic experimental analysis of which techniques actually improve the probability-based rankings, and by how much. In this paper we first discuss why the decision-tree representation is not intrinsically inadequate for probability estimation. Inaccurate probabilities are partially the result of decision-tree induction algorithms that focus on maximizing classification accuracy and minimizing tree size (for example via reduced-error pruning). Larger trees can be better for probability estimation, even if the extra size is superfluous for accuracy maximization. We then present the results of a comprehensive set of experiments, testing some straightforward methods for improving probability-based rankings. We show that using a simple, common smoothing method—the Laplace correction—uniformly improves probability-based rankings. In addition, bagging substantially improves the rankings, and is even more effective for this purpose than for improving accuracy. We conclude that PETs, with these simple modifications, should be considered when rankings based on class-membership probability are required.

Keywords: ranking, probability estimation, classification, cost-sensitive learning, decision trees, Laplace correction, bagging

1. Introduction

Tree-induction programs have received a great deal of attention over the past fifteen years in the fields of machine learning and data mining. Several factors contribute to their popularity. Tree-induction programs are fast and effective (Lim, Loh, & Shih, 2000). They work remarkably well with no tweaking of parameters, which has facilitated their wide use in the comparison of different learning algorithms. Tree induction also works comparatively well with very large data sets (Provost & Kolluri, 1999), with large numbers of variables, and with mixed-type data (continuous, nominal, Boolean, etc.). These qualities result in part from the simple yet powerful divide-and-conquer algorithm underlying tree learners, and in part from the high-quality software packages that have been available for learning decision trees (most notably, CART (Breiman et al., 1984) and C4.5 (Quinlan, 1993)).

As they have been used in most research and applications, tree induction programs produce classifiers (we do not consider regression here). These are models that map instances described by a vector of independent variables to one of a set of classes. However, as described below, in many applications this is not sufficient; a ranking based on the probability of class membership is needed, for example, so that a person can consider first the cases most likely to belong to the class. As we will show, the model that produces the best classifications does not necessarily produce the best probability-based rankings.

Because of the attractive properties of tree induction, probability estimation trees (PETs)—trees that estimate the probability of class membership—are seeing increasing use in such applications. Unfortunately, trees have been observed to produce poor estimates of class probabilities (Breiman, 1998, 2000; Pazzani et al., 1994; Smyth, Gray, & Fayyad, 1995; Bradley, 1997; Provost, Fawcett, & Kohavi, 1998). Several researchers have proposed techniques to improve the estimates, yet to our knowledge there has not been a systematic study of their efficacy for ranking.

In this paper, we present a study of how well these techniques improve the quality of rankings based on estimated class-membership probability. We first discuss prior work using and improving probability estimation trees. We then show that the decision tree *representation* is not inherently doomed to produce poor estimates, and that part of the problem is that modern decision-tree induction algorithms are biased against building accurate PETs. We use the results of this analysis and the suggestions of prior work to make a number of simple modifications to the popular decision-tree learning program C4.5. We apply the first pair of modifications to some simple synthetic problems, demonstrating the improvement in the probability estimates. We then report the results of a comprehensive experiment in which several modifications are applied to a wide variety of benchmark data sets. The results provide strong evidence that it is indeed possible to improve substantially the quality of probability-based ranking models produced by tree induction.

2. Prior work

PETs recently have seen increasing use by practitioners and researchers, for example in speech recognition (Jelinek, 1997), as node models in Bayesian networks (Friedman & Goldszmidt, 1996), in the recently introduced dependency-network representation and its application to collaborative filtering and other areas (Heckerman et al., 2000), in network diagnosis (Danyluk & Provost, 2002), and in cost-sensitive learning research (Domingos, 1999; Provost, Fawcett, & Kohavi, 1998). As described above, tree induction has many attractive properties. Under what conditions would it be desirable or necessary for a learned tree to produce effective probability-based rankings?

In many situations, rankings are more appropriate than categorical predictions. For example, a news-story filter or a web-page recommender may use the probability that an instance is a member of the class "interesting to user" to rank previously unseen instances for presentation. A fraud detection system may need to rank accounts by the probability that they have been compromised.

How are probability estimates typically generated from decision trees? Recall that tree induction partitions a data set recursively at each node. Each leaf (terminal node) defines

the subset of the data corresponding to the conjunction of the conditions along the path back to the root. The goal of the decision-tree learning program is to make these subsets be less "impure," in terms of the mixture of class labels, than the unpartitioned data set. For example, consider an unpartitioned population with two equally represented classes (maximally impure). A leaf node defining a subset of the population of which 90% are one class would be much less impure, and may facilitate accurate classification (only 10% error if this subset were classified as the majority class).

The previous example illustrates how class-membership probabilities typically are generated from decision trees. If a leaf node defines a subset of 100 training instances, 90 of which are one class (call it the "positive" class), then in use, any instance that corresponds to this leaf is assigned a probability of 0.9 (90/100) that it belongs to the positive class.

Notice a potential problem with this method of probability estimation. What if a leaf comprises only 5 training instances, all of which are of the positive class? Are you willing to have your probability estimator give an estimate of 1.0 (5/5) that subsequent instances matching the leaf's conditions also will be positive? Perhaps 5 instances is not enough evidence for such a strong statement? There are two potential direct solutions to this problem. One is that a statement of confidence in the probability estimation accompany the estimate itself; then decision making could take the confidence into account (Apte et al., 1999). The second potential solution is to "smooth" the probability estimate, replacing it with a less extreme value. We consider only the latter in this paper.

Smoothing of probability estimates from small samples is a well-studied statistical problem (Simonoff, 1995), and we believe that a thorough study of what are the best methods (and why) for PETs would be a useful contribution to machine-learning research. In this paper we focus on the method that has become a de facto standard for practitioners: the so-called Laplace estimate or Laplace correction. Assume there are k examples of the class in question at a leaf, N total examples, and C total classes. The maximum-likelihood estimate presented above calculates the estimated probability as $\frac{k}{N}$. The Laplace estimate calculates the estimated probability as $\frac{k+1}{N+C}$. Thus, while the frequency estimate yields a probability of 1.0 from the $k = 5$, $N = 5$ leaf, for a two-class problem the Laplace estimate yields a probability of $\frac{5+1}{5+2} = 0.86$. The Laplace correction can be viewed as a form of Bayesian estimation of the expected parameters of a multinomial distribution using a Dirichlet prior (Good, 1965; Buntine, 1991). It effectively incorporates a prior probability of $\frac{1}{C}$ for each class—note that with zero examples the estimated probability of each class is $\frac{1}{C}$. This may or may not be desirable for a specific problem; however, practitioners have found the Laplace correction worthwhile. To our knowledge, the Laplace correction was first introduced in machine learning by Niblett (1987). Clark and Boswell (1991) incorporated it into the CN2 rule learner, and its use is now widespread. For decision-tree learning the Laplace correction[1] has been used by certain researchers and practitioners (Pazzani et al., 1994; Bradford et al., 1998; Provost, Fawcett, & Kohavi, 1998; Bauer & Kohavi, 1999; Danyluk & Provost, 2002), but others still use maximum-likelihood estimates.

A more complex method for producing class probability estimates from decision trees is described by Smyth, Gray, and Fayyad (1995). They do not concentrate on the smaller leaves, as we have in the discussion so far. Instead they suggest a problem with estimating probabilities from the larger leaves. Specifically, they note that every example from a

particular leaf will receive the same probability estimate. They question whether the coarse granularity of these probability estimates may lead to reduced accuracy. To address this problem, they make a fundamental change to the representation. Specifically, at each leaf of the decision tree they place a kernel-based probability density estimator (just for the subset of the population defined by the leaf). They show that this method produces substantially better probability estimates than standard decision-tree programs (CART and C4.5).

This approach seems well founded and quite promising, but it does not address the question of whether there is a fundamental problem with using decision trees for probability estimation. If in fact there is, then showing that the new method outperforms the probability estimates of CART and C4.5 is not particularly informative. Therefore it is important to investigate whether simple modifications can improve the probability estimates of standard tree induction.

Finally, we should note that simply producing a probability estimate may not be enough for a real-world application. In a recent application of data mining techniques (including decision trees) to estimate probabilities for discovering insurance risk, Apte et al. (1999) describe in detail a variety of complications that also must be considered. For this paper, all we address is the production of probability estimates in order to produce rankings.

3. Representation versus induction

Viewed as probability estimators, trees consist of piecewise uniform approximations within regions defined by axis-parallel boundaries. Intuitively this may not seem as appropriate as a numeric method that estimates class probabilities as smoothly varying continuous outputs. However, trees *in principle* can be fine probability estimators. To see this we first must separate trees as a representation from the tree induction algorithm. Here we will consider the former. In the next section we will see that problems arise with the latter.

First consider nominal attributes. The tree represents the relevant combinations of features—relevant conditional probabilities. Any discrete conditional probability distribution can be represented by a PET.

For continuous attributes, a sufficiently large PET can estimate any class probability function to arbitrary precision. Consider the simple univariate, two-class problem depicted in figure 1: each class is distributed normally about a different mean. These overlapping probability densities define a continuous class-membership probability function over the domain of the variable (call it x). This may be one of the worst possible problems to which to apply a PET, because piecewise-uniform representations are obviously a poor inductive bias, and moreover because the problem is easy for other sorts of density estimators. However, for this and for any such problem a PET *can* estimate the probability of class membership to arbitrary precision. For this problem, each split in the tree partitions the x-axis, and each leaf is a segment of the x-axis. A PET would estimate the probability by looking at the class distribution for its segment (which in the figure can be seen by cutting a vertical slice and looking at the relative heights of the curves of the two classes in the slice). The key is to note that as the number of leaves increases, the slices become narrower, and the probability estimates can become more and more precise. In the limit, the tree predicts class probability perfectly.[2]

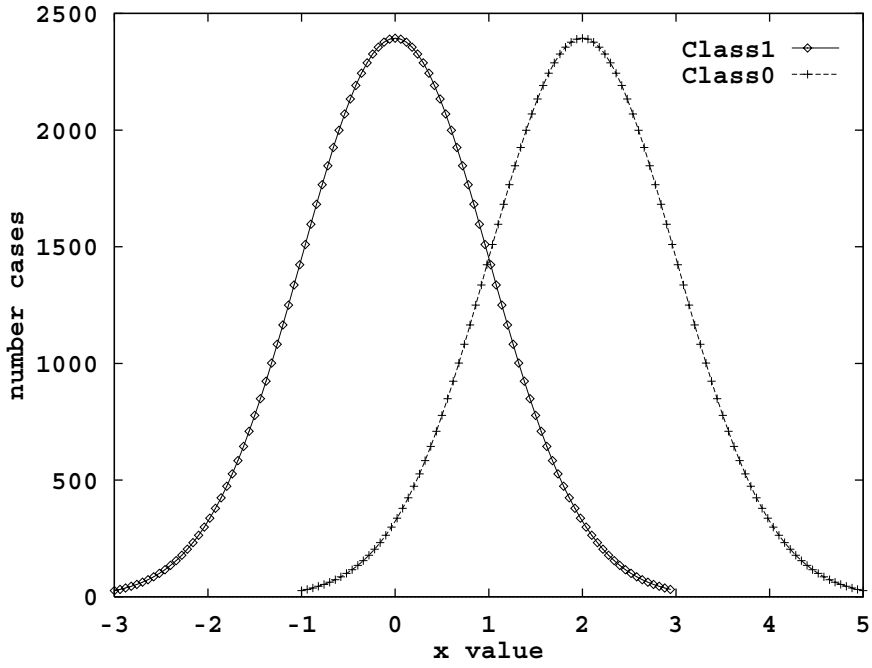

Figure 1. A test problem: Overlapping Gaussians.

Of course, *learning* such PETs is our ultimate interest. In the case of figure 1, other methods would learn better using fewer examples. But when the dimensionality of the problem is even moderately high, and little is known about the form of the underlying distribution, a piecewise-uniform approximation may well have lower bias or variance or both than smoother estimators.

4. Why PETs behave badly

So the question remains: why is it observed repeatedly that the decision trees produced by standard algorithms do not yield good probability estimates?

The answer is in the tree-building algorithm, not in the representation. For a historical perspective, it is useful to take a higher-level view of the research focus that (in part) drove much work on building decision trees. Decision trees have been evaluated, for the most part, by two criteria: classification accuracy and tree size (smaller is better). These have led to a wide variety of heuristics that have been remarkably successful at building small, accurate decision trees. However, *these very heuristics reduce the quality of the probability estimates.*

Why? Consider again our problem of univariate, overlapping Gaussians. What is the smallest, accuracy-maximizing decision tree? It is the tree with a single split at $x = 1$. This

separates the classes as well as any decision tree, and among the accuracy-maximizing trees it has minimal size. Thus, a good decision-tree building algorithm should return this simple tree (or a close approximation thereto). However, this tree's class probability estimates are not very accurate. All data points on one side of the split are assigned the same probability, corresponding to the proportion of the class that falls on the corresponding side of the split.

Above we say that this behavior (pathological from the PET point of view) is due to the tree-building algorithm, but we can be more specific. Modern decision-tree building algorithms first grow a (sometimes very) large tree, and then *prune* it back. The pruning stage tries to find a small, high-accuracy tree. Various pruning strategies are used. One such strategy is reduced-error pruning: remove sub-trees if they seem not to improve resultant accuracy on a validation set. In our example above, if the first split is correct, no subtree will improve accuracy. We believe that the details of the growing phase are less critical to obtaining good PETs than the choice of pruning mechanism. In particular, the commonly used splitting criteria (e.g., information gain and Gini index) also appear reasonable when the goal is to obtain good probability-based rankings. This is reinforced by the observations of Breiman et al. (1984) and Drummond and Holte (2000) that misclassification-cost effectiveness generally is insensitive to the choice of splitting criteria.

5. Training well-behaved PETs

Our question is whether we can build trees that yield better class probability estimates. The foregoing analysis suggests that pruning is the culprit. Looking more closely, we see that pruning removes two types of distinctions made by the tree: (i) false distinctions—those that were found simply because of "overfitting" idiosyncrasies of the training data set, and (ii) distinctions that indeed generalize (e.g., entropy in fact is reduced), and in fact will improve class probability estimation, but do not improve accuracy.

5.1. C4.4

To build better PETs we would like not to prune away distinctions of the latter type. The simplest strategy for keeping type-ii distinctions is simply not to prune at all. We can see on our overlapping-Gaussians problem that this strategy indeed gives us the desired result. In particular, we modified C4.5 by turning off pruning, turning off "collapsing" (a little-known pruning strategy that C4.5 performs even when growing its "unpruned" tree), and calculating class probabilities with the Laplace correction. We call this version C4.4.[3]

On the overlapping Gaussians problem with 100,000 training examples, C4.5 with pruning was used to build a PET (using the Laplace correction at the leaves), as was C4.4 (no pruning, no collapsing, Laplace correction). Figure 2 shows the performance of the PETs learned by C4.5 and C4.4. The solid line represents the true class probability boundary of the overlapping Gaussians problem (from figure 1). The class probability estimates given by C4.5 and C4.4 produce a piecewise-constant function, as expected. Note that C4.5 indeed finds a high-accuracy split, but the probability estimates (the horizontal segments) do not track the true class probability boundary well at all. C4.4's PET tracks the class probability boundary remarkably well.

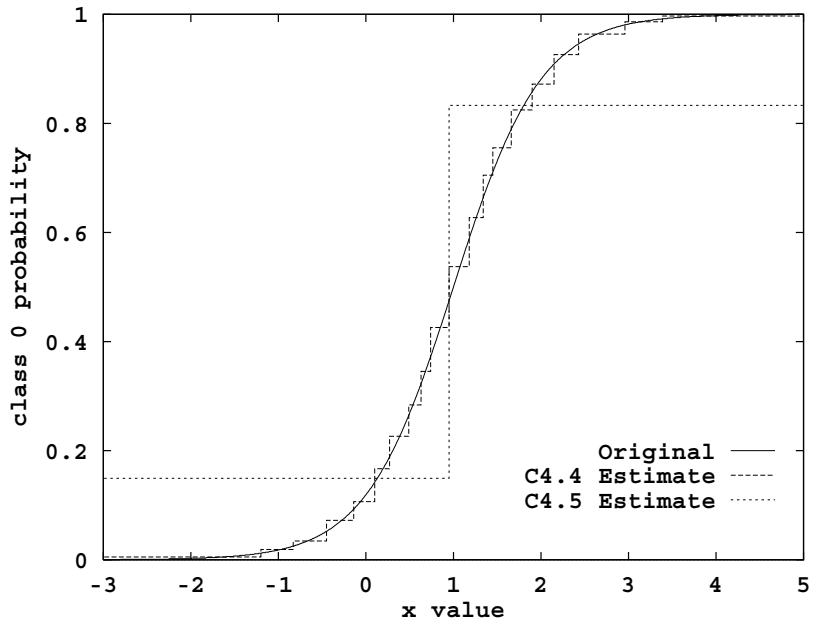

Figure 2. Class probability estimates for C4.5 and C4.4 on the overlapping Gaussians problem.

Of course, one may argue that the boundary still is rather rough,[4] and that an estimate with a better bias (e.g., a sigmoid function of the input) would perform better. As we mentioned earlier, the univariate, overlapping-Gaussians problem is one of the worst possible applications for a PET, in part because it is easy to propose a better alternative. However, consider the class probability function shown in figure 3. This will be more difficult for most methods than the problem in figure 2.

Now, consider the performance of C4.5 versus C4.4 on this problem. Note once again that for this probability function, the optimal decision tree also is a single cut, this time at a point in the interval $(-1,0)$. Therefore, the following should be viewed simply as a demonstration of the potential power of PETs over decision trees.

Once again, C4.5 with pruning was used to build a PET (using the Laplace correction at the leaves), as was C4.4 (no pruning, no collapsing, Laplace correction) from 100,000 training examples. The class probability borders learned by C4.5 and by C4.4 are shown in figure 4. As before, and as expected, C4.5 places a single split very near to the point where error should be minimized. Of course, this gives poor probability estimates for almost all instances. C4.4, on the other hand, produces class probability estimates that track the actual class probability border quite well.

5.2. Probability-bagging

In the foregoing, we assumed that the goal was to improve the probability estimates resulting from a single tree. A different strategy for using tree induction for probability estimation

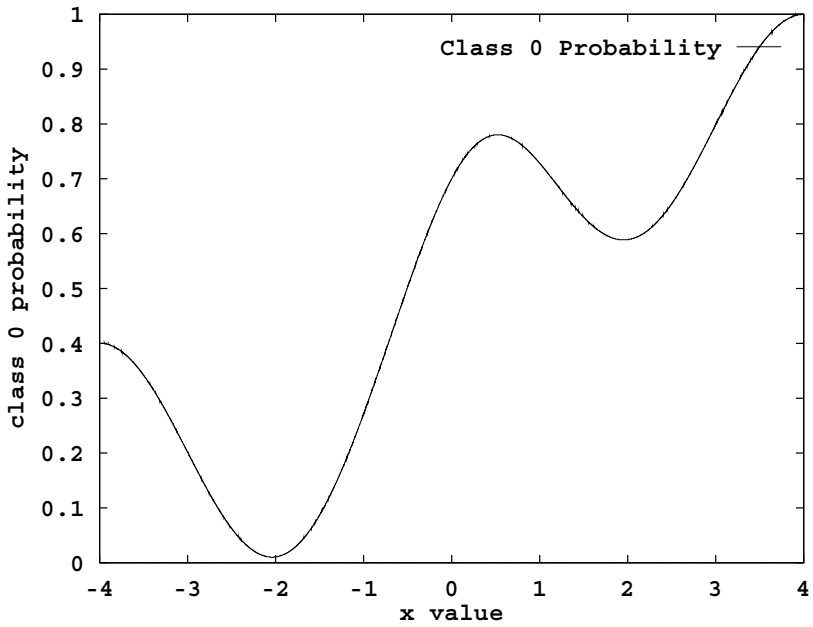

Figure 3. A more complex class probability function.

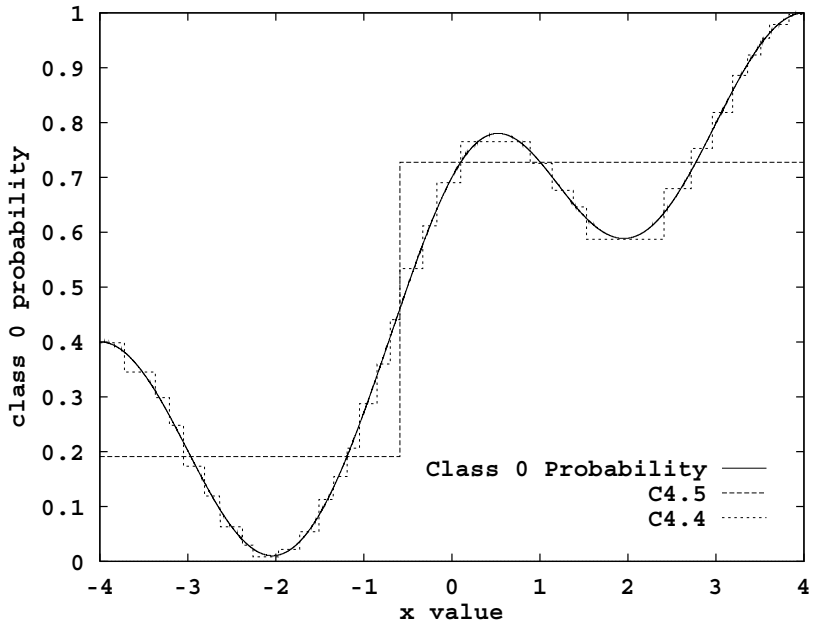

Figure 4. Learned probability borders: 100,000 training examples.

has received attention recently. Ensembles of classifiers, which learn multiple classification models and then combine their predictions (e.g., having them vote on a classification), have recently been shown often to improve classification accuracy when compared to using a single model. For example, bagging (Breiman, 1996) has been shown to outperform single model techniques with surprising consistency.

Recent results suggest that the improvements from bagging also apply to the use of trees for probability estimation and ranking, when probability estimates are averaged across the members of the ensemble (Provost, Fawcett, & Kohavi, 1998; Bauer & Kohavi, 1999). We should note that averaging multiple trees to produce probability estimates is not a novel product of the recent interest in multiple models; Buntine studied the technique a decade ago (Buntine, 1991). However, experiments have led us to the conclusion that bagging and the Bayesian averaging studied by Buntine are in fact quite different (Domingos, 1997). We include probability-bagging of PETs in our experimental comparison.

6. Comparing PETs

For this paper, we are interested in how well the learned models can rank new cases by the probability of class membership. The standard comparison method in machine learning research, comparing undifferentiated error rates, is not appropriate (Provost, Fawcett, & Kohavi, 1998), because it only assesses to what extent the estimated probabilities are on the correct side of the classification threshold (normally 0.5). One alternative is to use full-fledged ROC analysis (Swets, 1988), which compares visually a ranking's quality across the entire range of possible classification thresholds. As described in detail by Provost and Fawcett (2001), an ROC curve is generated from a ranking model as follows. The examples in the test set are ranked by the scores given by the model. If there are S different scores, there are $S + 1$ thresholding ranges, each of which will produce different classification performance (as can be characterized by the true-positive and false-positive rates) on these test data. Provost and Fawcett (1997, 2001) describe how for any two-class problem, precise, objective comparisons can be made with ROC analysis for various (and even unknown) conditions, such as different misclassification costs, different marginal class distributions in the target environment, different target classification thresholds, etc.

However, for the purpose of this study, we want to evaluate the probabilities generally rather than under specific conditions or under ranges of conditions. A subtle issue arises when evaluating the quality of the probabilities in our setting: although the trees are estimating probabilities of class membership, neither for the training data nor for the test data do we know the true probabilities. All we know is the true class of each example. For this paper, our task is simplified because all we address is how well the estimated probabilities rank cases (by the likelihood of class membership).

Knowing nothing about the task for which they will be used, which probabilities are generally better for ranking cases? The Wilcoxon-Mann-Whitney non-parametric test statistic ("the Wilcoxon") is appropriate for this comparison (Hand, 1997). The Wilcoxon measures, for a particular model, the probability that a randomly chosen class-0 case will be assigned a higher class-0 probability than a randomly chosen class-1 case. Therefore

a higher Wilcoxon score indicates that the probability-based ranking is better *generally* (there may be specific conditions under which the classifier with a lower Wilcoxon score is preferable). Importantly, for the purposes of this paper the calibration of the probabilities is not important, if the estimates rank well.[5] Another metric for comparing classifiers across a wide range of conditions is the area under the ROC curve (AUC) (Bradley, 1997); AUC also measures the quality of an estimator's ranking performance. Interestingly, it has been shown that the AUC is equivalent to the Wilcoxon statistic (Hanley & McNeil, 1982). (It also is essentially equivalent to the Gini coefficient (Hand, 1997).) Therefore, for this work we will report the AUC when comparing probability-based rankings. (Hand (1997) provides a thorough treatment of the comparison of class probability estimates both when the true probability distribution is known and when it is unknown.)

We examine whether, by making the modifications we make, the probability-based rankings generally improve. We make no claims as to whether one algorithm is "better" than another for the particular problems from which these data were drawn. The AUC measures judge the relative quality of the entire rankings.

To our knowledge, there previously has not been a systematic study comparing the performance of these PET variants for producing probability-based rankings. There do exist two closely related studies, that partially motivate the current study.

Bauer and Kohavi (1999) compare across 14 UCI data sets the quality of the probability estimates produced by PETs based on MC4 (their implementation of C4.5), a Laplace-corrected version of MC4 (using the m-estimate Laplace correction), and probability-bagging of MC4. They compare a mean-squared error measure of the quality of the probability estimates, computed as the square of one[6] minus the predicted probability of the correct class, averaged over the entire test set (we will call this measure 0/1-MSE). For these experiments they only report averages across the data sets, but their results are positive. They show a decrease in the average 0/1-MSE from 10.7% for unpruned C4.5 to 10.0% for Laplace-corrected unpruned C4.5 to 7.5% for probability-bagged C4.5.

Provost, Fawcett, and Kohavi (1998) compared the rankings of some of these PET variants. Specifically, they present the ROC curves of six algorithms evaluated on ten data sets, including Laplace-corrected PETs and probability-bagged PETs. They do not discuss which algorithms are better (this was not the purpose of the paper), but one can observe in their graphs that the ROC curves for probability-bagged PETs have larger areas that the curves for the PETs. In fact, in all but one case, the probability-bagged PETs completely dominate the curves of individual PETs. Our results, below, clarify and extend these results by examining the differences carefully, and by extending the study to a large number of data sets and to multiple-class problems.

Note that an improvement in 0/1-MSE does not necessarily indicate better probability-based rankings. In fact, a perfect ranking can have a worse 0/1-MSE than a ranking with an error in the first position. This is not the case for AUC. Also, we would like to see *how often* these techniques lead to improvements. Therefore, we will look individually at a larger number of domains.

7. Experiments and results

7.1. Methodology and results

We used the following 25 databases from the UCI repository (Blake & Merz, 2000): audiology, breast cancer (Ljubljana), chess (king-rook vs. king-pawn), credit (Australian), diabetes, echocardiogram, glass, heart disease (Cleveland), hepatitis, hypothyroid, iris, LED, liver disorders, lung cancer, lymphography, mushroom, primary tumor, promoters, solar flare, sonar, soybean (small), splice junctions, voting records, wine, and zoology. Each database was randomly divided 20 times into 2/3 of the examples for training and 1/3 for testing. The results presented are averages of these 20 runs. For data sets with more than two classes we computed the expected AUC, which is the weighted average of the AUCs obtained taking each class as the reference class in turn (i.e., making it class 0 and all other classes class 1).[7] The weight of a class's AUC is the class's frequency in the data. The results obtained are shown in Table 1, and summarized in Table 2. "Sign test" is the significance level of a binomial sign test on the number of wins (with a tie counting as half a win; the normal approximation to the binomial was used). "Wilcoxon test" is the significance level of a Wilcoxon signed-ranks test. Our observations are summarized below.

7.2. Laplace correction and pruning

C4.4 is a marked improvement over C4.5. Most of this improvement is due to the use of the Laplace correction, which, despite its simplicity, is quite effective in improving the quality of a tree's probability estimates. Our results in this respect agree with, but are stronger than, the results of Bauer and Kohavi (1999). The uniformity of success of the simple Laplace correction (e.g., 21 wins, 2 ties and 2 losses vs. C4.5) is remarkable.

Not pruning (C4.4) outperforms pruning (C4.5L) in more databases than the reverse, but the difference is not significant. We hypothesize that these inconclusive results are due to two competing effects: when pruning is disabled, more leaves are produced, which leads to a finer approximation to the true class probability function, but there are fewer data points within each leaf, which increases the variance in the approximation. Which of these two effects will prevail may depend on the size of the database. The limited range of data-set sizes used in the experiments and the presence of many confounding factors preclude finding a clear pattern in our results. We hypothesize that as we move to larger and larger data sets, as seems to be the trend in data mining, the advantage of C4.4 will become stronger.

7.3. Probability-bagging

Bagging also substantially improves the quality of probability estimates in almost all domains, and the improvements are often very large. This also agrees with the results of Bauer and Kohavi using 0/1-MSE (Bauer & Kohavi, 1999). The present results also show, over the twenty-five data sets, *not a single case* where bagging degrades the probability estimates, as measured by AUC. This accords with results that can be inferred from the ROC curves shown by Provost, Fawcett, and Kohavi (1998) (as described above).

Table 1. Experimental results: Expected AUC (area under the ROC curve, as percentage of maximum possible) and its standard deviation for C4.5, C4.5 with the Laplace correction (C4.5-L), C4.4, probability bagged C4.5 (C4.5-B) and bagged C4.4 (C4.4-B).

Database	C4.5	C4.5-L	C4.4	C4.5-B	C4.4-B
Audiology	89.4 ± 0.8	91.1 ± 0.9	91.0 ± 0.8	94.7 ± 0.5	95.2 ± 0.6
Breast	60.9 ± 1.7	63.1 ± 1.4	60.6 ± 1.2	68.9 ± 1.3	67.4 ± 1.3
Chess	99.7 ± 0.1	99.7 ± 0.0	99.9 ± 0.0	99.9 ± 0.0	99.9 ± 0.0
Credit	87.9 ± 0.7	89.9 ± 0.5	87.3 ± 0.4	92.6 ± 0.5	92.1 ± 0.4
Diabetes	74.8 ± 0.9	76.9 ± 0.8	77.3 ± 0.7	83.4 ± 0.5	83.2 ± 0.5
Echocardio	54.1 ± 1.3	55.9 ± 1.6	57.7 ± 1.1	67.4 ± 1.5	67.8 ± 1.6
Glass	79.2 ± 0.9	81.3 ± 1.0	81.3 ± 0.8	88.9 ± 0.8	88.7 ± 0.8
Heart	76.0 ± 1.2	81.1 ± 1.1	83.6 ± 0.8	88.4 ± 0.6	89.1 ± 0.6
Hepatitis	64.3 ± 2.5	68.4 ± 2.2	76.7 ± 1.5	83.2 ± 1.4	84.0 ± 1.4
Iris	96.0 ± 0.6	96.9 ± 0.3	97.3 ± 0.4	99.0 ± 0.2	99.2 ± 0.2
LED	81.4 ± 0.9	81.9 ± 1.0	84.3 ± 1.0	90.6 ± 0.8	90.6 ± 0.9
Liver	62.6 ± 1.2	63.7 ± 1.1	64.8 ± 1.5	74.0 ± 0.7	73.9 ± 0.7
Lung	54.6 ± 3.6	51.1 ± 3.5	50.5 ± 3.3	65.3 ± 3.0	62.0 ± 3.4
Lympho	79.7 ± 1.4	83.0 ± 1.5	84.7 ± 0.8	91.2 ± 0.8	91.3 ± 0.8
Mushroom	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0	100.0 ± 0.0
Promoters	78.4 ± 1.6	82.9 ± 1.5	81.2 ± 1.5	93.0 ± 1.2	93.8 ± 1.0
Solar	87.5 ± 0.6	88.9 ± 0.5	88.6 ± 0.5	89.8 ± 0.5	89.7 ± 0.5
Sonar	70.5 ± 1.3	76.2 ± 1.4	76.5 ± 1.4	85.2 ± 1.4	84.5 ± 1.3
Soybean	98.2 ± 0.5	97.8 ± 0.7	97.8 ± 0.7	100.0 ± 0.0	100.0 ± 0.0
Splice	96.4 ± 0.2	97.7 ± 0.1	97.8 ± 0.1	98.7 ± 0.1	98.9 ± 0.1
Thyroid	94.4 ± 0.9	96.2 ± 0.5	97.0 ± 0.4	97.5 ± 0.4	98.6 ± 0.3
Tumor	68.8 ± 0.7	71.7 ± 0.7	68.5 ± 0.8	77.0 ± 0.7	76.0 ± 0.6
Voting	97.1 ± 0.4	98.2 ± 0.2	94.6 ± 0.7	98.6 ± 0.2	98.9 ± 0.1
Wine	94.3 ± 0.6	94.5 ± 0.7	94.4 ± 0.8	99.4 ± 0.1	99.4 ± 0.1
Zoology	96.4 ± 0.5	98.0 ± 0.4	98.4 ± 0.4	99.4 ± 0.3	99.6 ± 0.1

Table 2. Summary of experimental results: AUC comparisons.

Systems	Wins	Ties	Losses	Avg. diff. (%)	Sign test	Wilcoxon test
C4.4 vs. C4.5	18	1	6	2.0	1.0	0.3
C4.4 vs. C4.5-L	13	3	9	0.2	30.0	30.0
C4.5-L vs. C4.5	21	2	2	1.7	0.1	0.1
C4.5-B vs. C4.5	24	1	0	7.3	0.1	0.1
C4.4-B vs. C4.4	23	2	0	5.3	0.1	0.1
C4.4-B vs. C4.5-B	11	5	9	−0.1	45.0	50.0

It is noteworthy that bagging's improvements in AUC are on average much larger than its improvements in accuracy (7.3% vs. 2.8% for C4.5), indicating that bagging may be even more effective for improving probability estimators than for improving classifiers. The improvements in AUC are larger on average for C4.5 than for C4.4, presumably because there is more room for improvement in C4.5. Once bagging is used, whether or not pruning and the Laplace correction are used makes little difference. Despite its effectiveness, bagging has the disadvantage that any comprehensibility of the single tree is lost. However, individual PETs can be very large, especially when pruning is not used, so they themselves may or may not be comprehensible. Bagging also carries greater computational cost. When high-quality estimation is the sole concern, bagging should clearly be used. When comprehensibility and/or computational cost are also important, a single C4.4 tree may be preferable, or a method like CMM (Domingos, 1997) (which produces a single-tree approximation of the ensemble) may be useful.

8. Conclusions and discussion

The poor performance of PETs built by conventional decision-tree learning programs can be explained by a combination of factors. First, as shown by the demonstrations on synthetic data, the heuristics used to build small accurate decision trees are biased strongly against building accurate PETs. Larger trees can work better for probability estimation.

The second factor explaining the poor performance of conventional PETs is that, when a purely frequency-based (unsmoothed) estimate is used, small leaves give poor probability estimates. This is the probability-estimation counterpart of the well-known "small disjuncts problem": in induced disjunctive class descriptions, small disjuncts are more error-prone (Holte, Acker, & Porter, 1989). While this is not surprising statistically, the uniformity and magnitude of the improvement given by the simple, easy-to-use, Laplace correction nevertheless is remarkable.

A third factor, which we have not investigated, is the calibration of the probability estimates. Recently, Margineantu and Dietterich (2001) have investigated the issue of the accuracy of the estimates versus the accuracy of the rankings, and show that PETs indeed produce surprisingly good rankings, even when the probability estimates themselves are questionable.

Another significant observation is that probability-bagged PETs produce excellent probability-based rankings. As with accuracy, bagging substantially improves PETs. Moreover, over the twenty-five data sets we tested, bagging never degrades the probability estimates. Furthermore, bagging improves probability estimates (as measured by AUC) even more than it improves classification accuracy. The extent of this is quite remarkable: in 9 of 25 domains bagging gives an absolute AUC improvement of more than 0.10. We strongly echo the conclusion of Bauer and Kohavi (1999) that for problems where probability estimation is required, one should seriously consider using probability-bagged PETs—especially in ill-defined or high-dimensional domains.

Bagged PETs also have implications for other areas of data mining and machine learning research. For example, the MetaCost algorithm (Domingos, 1999) uses a bagged PET as a

subprocedure for cost-sensitive learning. The quality of the probability estimates obtained in this way was an open question; our results partially validate the procedure used.

The purpose of this work was to study how the probability-based rankings obtained by tree induction could be improved. We believe that the results we have presented have given us a substantially better understanding. However, what we did not study here is how these PETs compare with other methods for estimating probabilities. In a working version of this paper (Provost & Domingos, 2000) we hypothesized that as long as there are many examples, PETs can compete with traditional methods for building class probability estimators. Recent work shows that indeed this is the case. Perlich, Provost, and Simonoff (2003) show that for large data sets, tree induction often produces probability-based rankings that are superior to those produced by logistic regression (a standard statistical method for estimating class-membership probability). They also characterize the type of domain for which each method is preferable. A direction for future work is to study the incorporation of more sophisticated methods for improving probability estimates (e.g., shrinkage (Bahl et al., 1989; Hastie & Pregibon, 1990; Jelinek, 1997; McCallum et al., 1998)).

Acknowledgments

Ronny Kohavi suggested the use of probability bagging for our 1998 study with Tom Fawcett. Doug Fisher and our anonymous reviewers made suggestions that improved the paper considerably. We thank Claudia Perlich, Maytal Saar-Tsechansky, and Jeff Simonoff for enlightening discussions about probability estimation and ranking, all those who have pointed us to related work, and the contributors to and the librarians of the UCI repository for facilitating experimental research in machine learning. This work was partly supported by IBM Faculty Awards to both authors, and by an NSF CAREER Award to the second author.

Notes

1. Including a generalization known as the *m*-estimate (Cestnik, 1990; Dzeroski, Cestnik, & Petrovski, 1993; Kohavi, Becker, & Sommerfield, 1997).
2. A similar result for regression trees has been formally demonstrated by Gordon and Olshen (1984).
3. Note that Bradford et al. (1998) show that cost-sensitive tree pruning is no better than simply not pruning at all, as long as the Laplace correction is used.
4. Note that C4.5 uses a minimum description length heuristic to reduce spurious splitting on numeric attributes, and because of this the leaves remain larger than they would without the heuristic.
5. An inherently good probability estimator can be skewed systematically, so that although the probabilities are not accurate, they still rank cases equivalently. This would be the case, for example, if the probabilities were squared. Such an estimator will receive a high Wilcoxon score. A higher Wilcoxon score indicates that, *with proper recalibration,* the probabilities of the estimator will be better. Probabilities can be recalibrated empirically (Sobehart et al., 2000; Zadrozny & Elkan, 2001; Bennett, 2000). In addition to describing new calibration methods, Bennett provides an in-depth discussion of calibration, including additional related work.
6. Recall that for these data we only know the true class of each example, not the true probability of class membership for the example's description.
7. This is a minor variant of the method proposed recently by Hand and Till (2001).

References

Apte, C., Grossman, E., Pednault, E., Rosen, B., Tipu, F., & White, B. (1999). Probabilistic estimation-based data mining for discovering insurance risks. *IEEE Intelligent Systems, 14*, 49–58.

Bahl, L. R., Brown, P. F., de Souza, P. V., & Mercer, R. L. (1989). A tree-based statistical language model for natural language speech recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 37:7*, 1001–1008.

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, *36*, 105–142.

Bennett, P. (2002). Using asymmetric distributions to improve classifier probabilities: A comparison of new and standard parametric methods. Technical report CMU-CS-02-126, School of Computer Science, Carnegie Mellon University.

Blake, C., & Merz, C. J. (2000). UCI repository of machine learning databases. Machine-readable data repository, Department of Information and Computer Science, University of California at Irvine, Irvine, CA. Available at http://www.ics.uci.edu/~mlearn/MLRepository.html.

Bradford, J. P., Kunz, C., Kohavi, R., Brunk, C., & Brodley, C. E. (1998). Pruning decision trees with misclassification costs. *Proceedings of the Tenth European Conference on Machine Learning* (pp. 131–136). Berlin: Springer Verlag.

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition, 30:7*, 1145–1159.

Breiman, L. (1996). Bagging predictors. *Machine Learning, 24*, 123–140.

Breiman, L. (1998). Out-of-bag estimation. Unpublished manuscript.

Breiman, L. (2000). Private communication.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and Regression Trees*. Wadsworth International Group.

Buntine, W. (1991). *A theory of learning classification rules*. Ph.D. thesis, School of Computer Science, University of Technology, Sydney, Australia.

Cestnik, B. (1990). Estimating probabilities: A crucial task in machine learning. *Proceedings of the Ninth European Conference on Artificial Intelligence* (pp. 147–149). Pitman.

Clark, P., & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. *Proceedings of the Sixth European Working Session on Learning* (pp. 151–163). Berlin: Springer.

Danyluk, A., & Provost, F. (2002). Telecommunications network diagnosis. In W. Kloesgen, & J. Zytkow (Eds.), *Handbook of Knowledge Discovery and Data Mining*, 897–902.

Domingos, P. (1997). Why does bagging work? A Bayesian account and its implications. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (pp. 155–158). Menlo Park, CA: AAAI Press.

Domingos, P. (1999). MetaCost: A general method for making classifiers cost-sensitive. *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 155–164). New York: ACM Press.

Domingos, P. (1997). Knowledge acquisition from examples via multiple models. In D. H. Fisher (Ed.), *Proceedings of the Fourteenth International Conference on Machine Learning (ICML-97)* (pp. 98–106). San Francisco, CA: Morgan Kaufmann.

Drummond, C., & Holte, R. (2000). Exploiting the cost (in)sensitivity of decision tree splitting criteria. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 239–246). San Francisco: Morgan Kaufmann.

Dzeroski, S., Cestnik, B., & Petrovski, I. (1993). Using the *m*-estimate in rule induction. *Journal of Computing and Information Technology, 1*, 37–46.

Friedman, N., & Goldszmidt, M. (1996). Learning Bayesian networks with local structure. *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence* (pp. 252–262). San Francisco: Morgan Kaufmann.

Good, I. J. (1965). *The Estimation of Probabilities: An Essay on Modern Bayesian Methods*. Cambridge, MA: MIT Press.

Gordon, L., & Olshen, R. A. (1984). Almost sure consistent nonparametric regression from recursive partitioning schemes. *Journal of Multivariate Analysis, 15*, 147–163.

Hand, D. J. (1997). *Construction and Assessment of Classification Rules*. Chichester: John Wiley and Sons.

Hand, D. J., & Till, R. J. (2001). A simple generalization of the area under the ROC curve for multiple class classification problems. *Machine Learning, 45:2*, 171–186.

Hanley, J. A., & McNeil, B. J. (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology, 143*, 29–36.

Hastie, T. J., & Pregibon, D. (1990). Shrinking trees. Technical report, AT&T Laboratories.

Heckerman, D., Chickering, M., Meek, C., Rounthwaite, R., & Kadie, C. (2000). Dependency networks for density estimation, collaborative filtering, and data visualization. *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. San Francisco: Morgan Kaufmann.

Holte, R., Acker, L., & Porter, B. (1989). Concept learning and the problem of small disjuncts. *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence* (pp. 813–818). San Francisco: Morgan Kaufmann.

Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. Cambridge, MA: MIT Press.

Kohavi, R., Becker, B., & Sommerfield, D. (1997). Improving simple Bayes. *The Ninth European Conference on Machine Learning* (pp. 78–87).

Lim, T.-J., Loh, W.-Y., & Shih, Y.-S. (2000). A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning, 40:3*, 203–228.

Margineantu, D. D., & Dietterich, T. G. (2001). Improved class probability estimates from decision tree models. In C. Holmes (Ed.), *Nonlinear Estimation and Classification*. The Mathematical Sciences Research Institute, University of California, Berkeley.

McCallum, A., Rosenfeld, R., Mitchell, T., & Ng, A. Y. (1998). Improving text classification by shrinkage in a hierarchy of classes. *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 359–367). San Francisco: Morgan Kaufmann.

Niblett, T. (1987). Constructing decision trees in noisy domains. *Proceedings of the Second European Working Session on Learning* (pp. 67–78). Wilmslow, England: Sigma Press.

Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1994). Reducing misclassification costs. *Proceedings of the Eleventh International Conference on Machine Learning* (pp. 217–225). San Francisco: Morgan Kaufmann.

Perlich, C., Provost, F., & Simonoff, J. S. (2003). Tree induction versus logistic regression: A learning-curve analysis. *Journal of Machine Learning Research*. (In press).

Provost, F., & Domingos, P. (2000). Well-trained PETs: Improving probability estimation trees. CeDER Working Paper #IS-00-04, Stern School of Business, New York University, NY 10012.

Provost, F., & Fawcett, T. (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)* (pp. 43–48). Menlo Park, CA: AAAI Press.

Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning, 42*, 203–231.

Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. *Proceedings of the Fifteenth International Conference on Machine Learning* (pp. 445–453). San Francisco: Morgan Kaufmann.

Provost, F., & Kolluri, V. (1999). A survey of methods for scaling up inductive algorithms. *Data Mining and Knowledge Discovery, 3:2*, 131–169.

Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. San Francisco: Morgan Kaufmann.

Simonoff, J. S. (1995). Smoothing categorical data. *Journal of Statistical Planning and Inference, 47*, 41–69.

Smyth, P., Gray, A., & Fayyad, U. (1995). Retrofitting decision tree classifiers using kernel density estimation. *Proceedings of the Twelfth International Conference on Machine Learning* (pp. 506–514). San Francisco: Morgan Kaufmann.

Sobehart, J. R., Stein, R. M., Mikityanskaya, V., & Li, L. (2000). Moody's public firm risk model: A hybrid approach to modeling short term default risk. Tech rep., Moody's Investors Service, Global Credit Research. Available: http://www.moodysqra.com/research/crm/53853.asp.

Swets, J. (1988). Measuring the accuracy of diagnostic systems. *Science, 240*, 1285–1293.

Zadrozny, B., & Elkan, C. (2001). Obtaining calibrated probability estimates from decision trees and naive Bayesian classifiers. In C. Brodley, & A. Danyluk (Eds.), *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 609–616). San Francisco: Morgan Kaufmann.

Received October 12, 2000
Revised June 5, 2002
Accepted June 6, 2002
Final manuscript June 18, 2002