

Well-Trained PETs: Improving Probability Estimation Trees

Foster Provost, New York University

Pedro Domingos, University of Washington

CDER WORKING PAPER #00-04-IS, STERN SCHOOL OF BUSINESS, NYU, NY, NY, 10012

October 3, 2000

Abstract:

Decision trees are one of the most effective and widely used classification methods. However, many applications require class probability estimates, and 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 estimates, 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 a variety of different methods for improving PETs. The results show, somewhat surprisingly, that alternative pruning methods do not improve the probabilities. In contrast, the experiments show that using a simple, common smoothing method—the Laplace correction—uniformly improves probability estimates. In addition, bagging substantially improves probability estimates, and is even more effective for this purpose than for improving accuracy. We conclude that PETs, with these simple modifications, should be considered when class probability estimates are required.

Contact author:

Professor Foster Provost

Information Systems Department

Stern School of Business

New York University

44 W. 4th St., New York, NY 10012-1126, USA

Tel: +1.212.998-0806

Fax: +1.212.995-4228

E-mail: fprovost@stern.nyu.edu

1 Introduction

Decision-tree learning programs have received a great deal of attention over the past fifteen years in the fields of machine learning and KDD. Several factors contribute to their popularity. Decision-tree learning 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. Decision trees also work 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 decision-tree learners, and in part from the high-quality software packages that have been available for learning decision trees (most notably, CART (Breiman, Friedman, Olshen, & Stone, 1984) and C4.5 (Quinlan, 1993)).

There is another reason why decision-tree learning programs are so popular. In many situations black-box models, or models where the reasons for decisions are hidden behind opaque mathematical formulae, are unacceptable to users. This may be true because a system is going to incorporate the models, and certain managers have responsibility for the system's behavior (and therefore must understand its inner workings). Or incomprehensible models may be unacceptable because the model is built as a stage in a knowledge discovery process, in which the goal is to induce comprehensible models for human consumption. Decision trees are easy for people to understand. Furthermore, they can be transformed easily into rule sets, which are even more comprehensible (Quinlan, 1993; Fürnkranz, 1999).

As they have been used in most research and applications, decision trees are categorical classifiers. They 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 categorical classification is not sufficient; class probabilities are needed. Because of the attractive properties of decision trees, probability estimation trees (PETs)—decision trees that estimate the probability of class membership—are seeing increasing use in such applications. Unfortunately, decision trees have been observed to produce poor estimates of class probabilities (Breiman, 1998, 2000; Pazzani, Merz, Murphy, Ali, Hume, & Brunk, 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.

In this paper, we present such a study. We first discuss prior work using and improving probability estimation trees. We then explain 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 of a variety of modifications applied to a wide variety of benchmark data sets. The results show conclusively that it indeed is possible to improve substantially the quality of probability estimation in decision trees.

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, Chickering, Meek, Rounthwaite, & Kadie, 2000), in network diagnosis (Danyluk & Provost, 2000), and in cost-sensitive learning research (Domingos, 1999; Provost et al., 1998). As described above, decision-tree learning has many attractive properties. Under what conditions would it be desirable or necessary for a learned decision tree to produce class probability estimates?

If misclassification costs or the marginal (prior) class distribution can not be specified precisely when the classifier is built, it is impossible to specify the appropriate classification task. Instead of categorical classifications, models should estimate the probability of membership in the various classes. Similarly, in some situations rankings are preferred to categorical classifications. 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. Learning and classifying in such situations is described in detail elsewhere (Provost & Fawcett, 2000).

How are probability estimates typically generated from decision trees? Recall that a decision tree partitions the data 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 probabilities are typically 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.

Now you might 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, Grossman, Pednault, Rosen, Tipu, & White, 1999). The second potential solution is to “smooth” the probability estimate, replacing it with a less extreme value. We only consider the latter in this paper, in order to keep the scope of the project narrow and focused on decision trees that give more accurate probability estimates.

Smoothing of probability estimates from small samples is a well-studied statistical problem (Simonoff, 1998), 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 p examples of the class in question at a leaf, N total examples, and C total classes. The frequency-based estimate presented above calculates the estimated probability as $\frac{p}{N}$. The Laplace estimate calculates the estimated probability as $\frac{p+1}{N+C}$. Thus, while the frequency estimate yields a probability of 1.0 from the $p = 5, N = 5$ 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 (Buntine, 1991). It effectively incorporates a prior probability of $\frac{1}{C}$ for each class—note that with zero examples the 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 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 has been used by certain researchers and practitioners (Pazzani et al., 1994; Bradford, Kunz, Kohavi, Brunk, & Brodley, 1998; Provost et al., 1998; Bauer & Kohavi, 1999; Danyluk & Provost, 2000), but others still use frequency-based estimates.

To our knowledge, the most detailed treatment of the production of class probability estimates from decision trees is reported by Smyth, Gray and Fayyad (Smyth et al., 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 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 from our perspective it is problematic. First of all, one of the primary advantages of the decision-tree representation is its simplicity and modularity. In particular, because comprehensibility is so important, decision trees often are preferable to *single* density estimators, even when the latter have slightly better accuracy.¹ The new model is a complicated combination of *many* density estimators (and indeed Smyth et al. note that one way to see the method is that the decision-tree learner is a feature selector for density estimation). Equally important is a different problem. This work 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 beats the probability estimates of CART and C4.5 is not particularly

¹We have observed this in more than one real-world application of machine learning techniques.

impressive. Therefore it is important to investigate whether standard decision trees can be made better probability estimators. We note however that if they can, then the method of Smyth et al. might be improved by grafting the density estimators onto the more accurate PET.²

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 accurate probability estimates.

3 Representation versus induction

Viewed as probability estimators, decision trees construct 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, decision trees *in principle* can be fine PETs. To see this we first must separate decision trees as a representation from the 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 decision tree represents the relevant combinations of features—relevant conditional probabilities. Any 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 just about the worse problem to which to apply a PET, because piecewise-uniform representations are obviously a poor inductive bias, and moreover because the problem is rather 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 decision tree partitions the x -axis,

²Or, of course, the new PET may improve probability estimates so much that little can be gained by grafting on the density estimators.

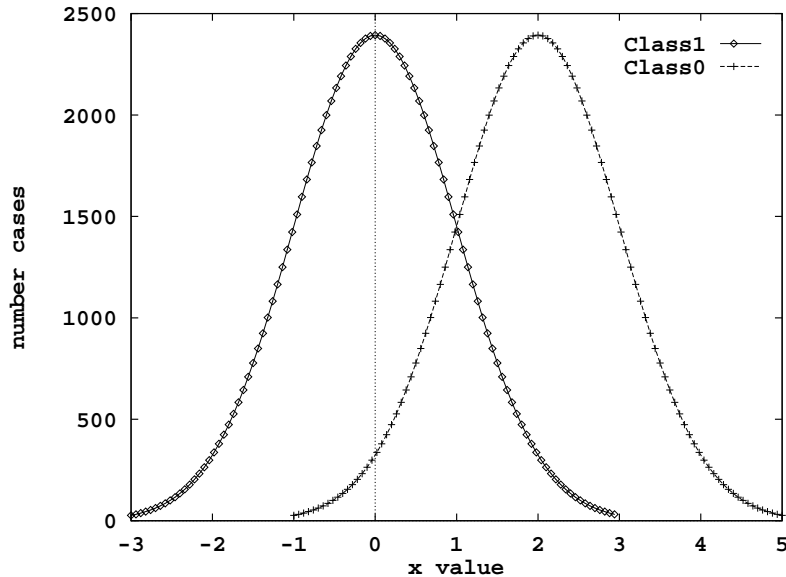


Figure 1: The test problem: Overlapping Gaussians.

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 decision tree predicts class probability perfectly.

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 and/or variance 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,

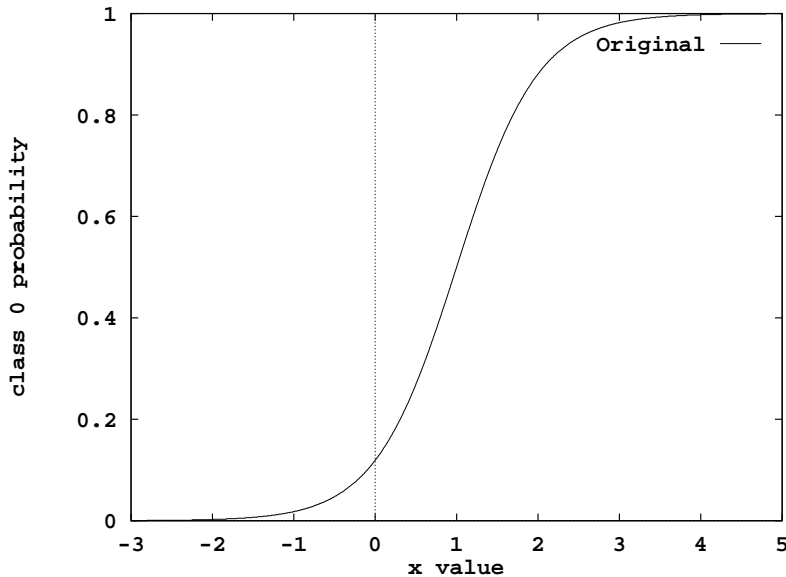


Figure 2: The corresponding class probability function.

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. So, a good decision-tree building algorithm should return this simple tree (or a close approximation thereto). But how good are this decision tree's class probability estimates? Not very good at all. All data points on one side of the split are assigned the same probability, e.g., the proportion of the class that fall 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 estimates. This is reinforced by the observations of Breiman et al. (1984) and Drummond and Holte (2000) that misclassification costs are generally 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 decision 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 (we will return to the former later). 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.

We hypothesized that C4.4 may beat C4.5 at probability estimation. Of course this went against our better intuition, established by years of reading machine learning papers touting the virtues of pruning. However, in the literature there are hints of support for such a hypothesis. For example, as mentioned above, Bradford et al. (Bradford et al., 1998) show that cost-sensitive decision-tree pruning is no better than simply not pruning at all, as long as the Laplace correction is used. One possible reason is that unpruned decision trees give very good probability estimates.³

Figure 2 shows the class probability boundary of the overlapping Gaussians problem (from Figure 1).

³If a model gives very good probability estimates, it inherently is cost sensitive (Provost & Fawcett, 1998).

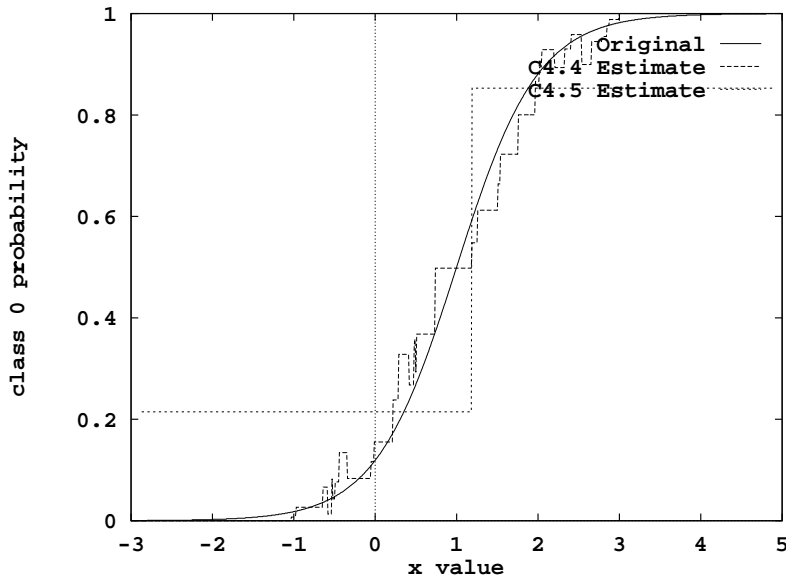


Figure 3: Comparing class probability estimates.

Figure 3 shows the performance of the PETs learned by C4.5 and C4.4 on the overlapping Gaussians problem. This was generated from trees built with 100,000 examples. The class probability estimates given by C4.5 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.

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 about the worst possible application for a PET, in part because it is easy to propose a better alternative. However, consider the class probability function shown in Figure 4. This will be more difficult for most methods than the problem in Figure 3.

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.

C4.5 with pruning was used to build a PET (using the Laplace correction at the leaves),

⁴Note 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.

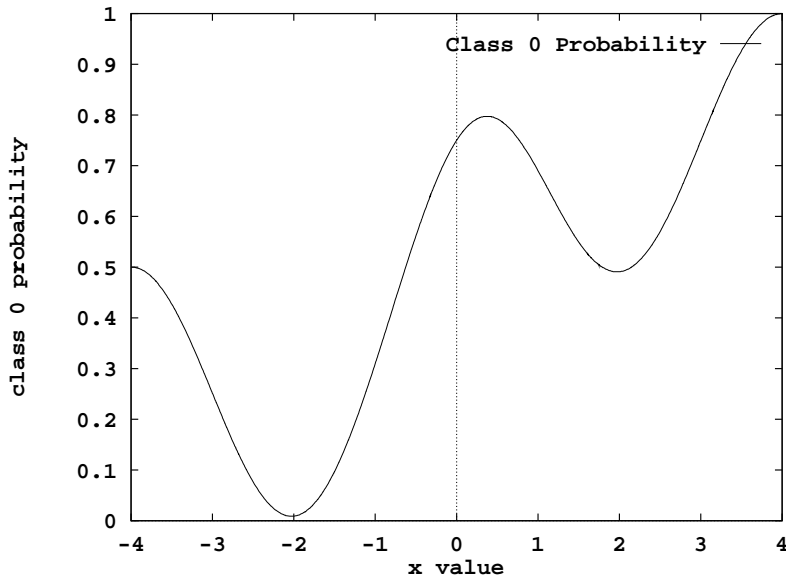


Figure 4: A more complex class probability function.

as was C4.4 (no pruning, no collapsing, Laplace correction). The class probability borders learned by C4.5 and by C4.4 are shown in Figure 5.

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. As more data are used to build the tree, the class probability estimates become more precise. Figure 5 shows the result of training the PETs on 10,000 training examples. Figure 6 shows the result of training the PETs on 100,000 training examples. Notice that as the training sets get larger, both C4.5 and C4.4 do better at their primary task. C4.5's single split is closer to the point where accuracy is maximized. C4.4 produces finer-grained probability estimates that track the actual border more precisely.

5.2 So where is the rub?

Of course, training PETs in practice is not that simple. As we mentioned earlier, there are two types of distinctions removed by pruning. In arguing for C4.4 we highlighted distinctions of type-ii, which obviously should be retained for probability estimation. However, we ignored distinctions of type-i: spurious distinctions resulting from overfitting the training set. In the previous sections C4.4 was applied to plenty of data, given the low dimensionality of the

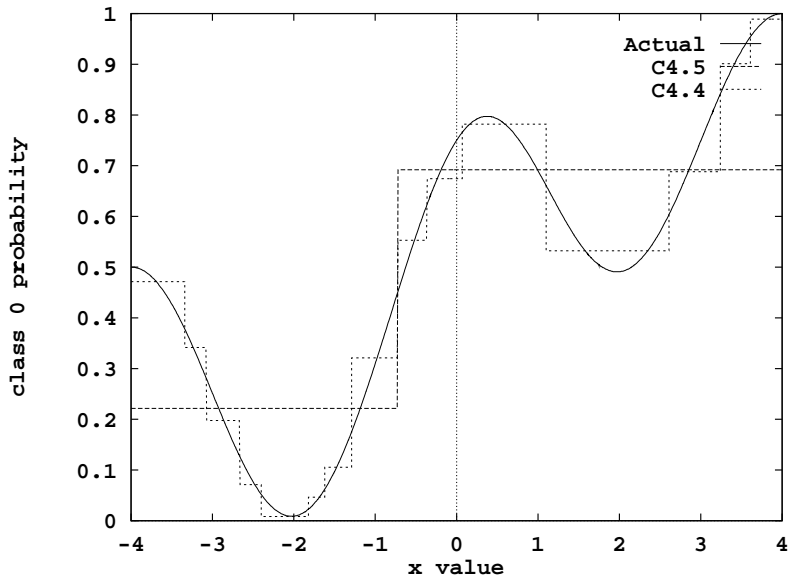


Figure 5: Learned probability borders: 10,000 training examples.

problem. What will happen when data is sparse? Will not C4.4 produce false distinctions that will distort its probability estimates?

It almost certainly will. Do the benefits of C4.4 outweigh the drawbacks? Are the PETs produced by C4.4 better than those produced by C4.5? We evaluate this empirically below. A further question is whether there is an effective middle ground. Pruning based on minimizing accuracy obviously is not the right thing to do. On the other hand, not pruning at all may be too drastic. It might be useful to prune with the specific goal of preserving distinctions that are important for probability estimation.

Reduced-error pruning is not the only pruning strategy that has been used in building decision trees. A strategy that seems better aligned with the goal of retaining distinctions that are significant from the perspective of probability estimation is chi-square pruning. With chi-square pruning, leaves of the tree are collapsed to their parent node if a chi-square test does not indicate that there is a significant difference in the class distributions before and after the split. Several decision-tree learning algorithms have used variations of chi-square pruning (Quinlan, 1986; Jensen & Schmill, 1997; Kass, 1980). Perhaps most notably, C4.5’s predecessor, ID3 (Quinlan, 1986) used chi-square “prepruning”; it stopped growing the tree when a chi-square test did not show a significant difference in the distributions.

We hypothesized that an augmented C4.4, using chi-square pruning, would yield improved performance over C4.4. Such a procedure would be parameterized by the p-level at

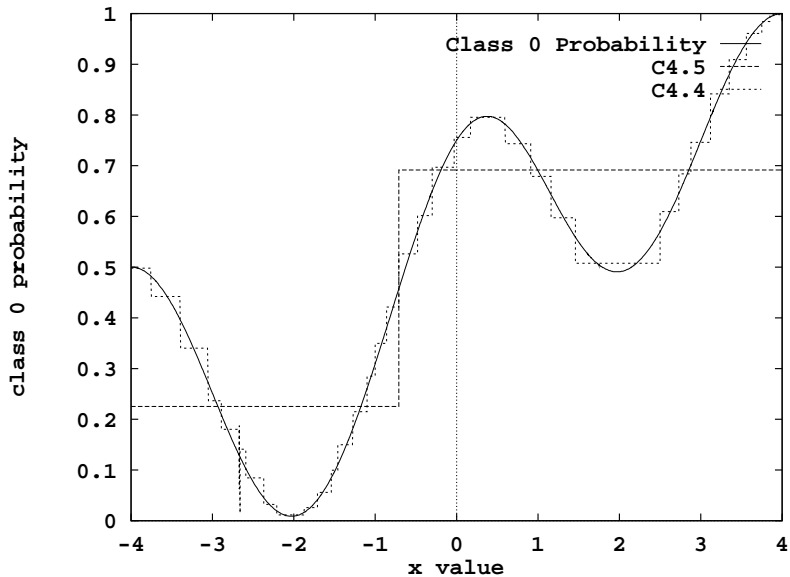


Figure 6: Learned probability borders: 100,000 training examples.

which pruning would occur, and the question then would arise as to how to set the p-level appropriately. However, given a sufficient amount of data, cross-validation could be used to determine empirically what p-level would be appropriate.

5.3 Another alternative for building PETs

In the foregoing, we assumed that the goal was to improve the probability estimates resulting from a single tree. A different strategy for using decision trees for probability estimation has received attention recently. Multiple-model 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 decision trees for probability estimation (Provost et al., 1998; Bauer & Kohavi, 1999). We should note that averaging multiple decision trees to produce probability estimates is not a novel product of the recent interest in multiple models; Buntine studied the technique ten years ago (Buntine, 1991). However, our experiments have led us to the conclusion that bagging and the Bayesian averaging studied by Buntine are in fact quite different (Domingos, 1997).

6 Experiments and Results

The results presented above were obtained from simple synthetic data. We were interested in whether the improved performance hypothesized for C4.4, and observed above, generalized to data from real-world problems. We also were interested in verifying or refuting our other hypothesized improvements, including chi-square pruning and bagging.

6.1 Comparison metric

For this work it is necessary to evaluate and compare different models with respect to their estimates of class probabilities. In the standard machine-learning evaluation paradigm, the true class probability distributions are not known. Instead, a set of instances is available, labeled with the true class. Comparisons are based on estimates of performance from these data.

The standard method, comparing undifferentiated error rates, is obviously not appropriate (Provost et al., 1998). One alternative is to use ROC analysis (Swets, 1988), which compares visually the classifiers' performance across the entire range of probabilities. Provost and Fawcett (Provost & Fawcett, 1997, 1998) describe how precise, objective comparisons can be made with ROC analysis.

However, for the purpose of this study, we want to evaluate the probabilities generally rather than under specific conditions or under ranges of conditions. Knowing nothing about the task for which they will be used, which probabilities are generally better? The Wilcoxon-Mann-Whitney non-parametric test statistic (the Wilcoxon) (Hand, 1997) is appropriate for this comparison. The Wilcoxon measures, for a particular classifier, 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 higher Wilcoxon score indicates that the probabilities are *generally* better (there may be specific conditions under which the classifier with a lower Wilcoxon score is preferable), if calibration of the probabilities is ignored.⁵ Another metric for comparing classifiers across a wide range of conditions is the area under the ROC curve

⁵An 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, for example as described by Soberhart et al. (2000).

(AUC) (Bradley, 1997); AUC measures the quality of an estimator’s classification performance, averaged across all possible probability thresholds. Interestingly, it has been shown that the AUC is equivalent to the Wilcoxon statistic (Hanley & McNeil, 1982). (It also is equivalent to the Gini coefficient (Hand, 1997).) Therefore, for this work we will report the AUC when comparing class probability estimators. (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 are interested in whether, by making the modifications we make, the probabilities generally improve. We make no claims as to whether one algorithm is “better” than another for the problems from which these data were drawn. The AUC metric(s) judge the relative quality of the probabilities averaged over all possible output thresholds. It may be the case that for a particular set of conditions under which the PETs will be used, i.e., where a particular output threshold is called for, a PET with a lower AUC score in fact is desirable.

6.2 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). 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.

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, C4.4 with chi-square pruning with a 5% significance threshold (C4.4-X), bagged C4.5 (C4.5-B) and bagged C4.4 (C4.4-B).

Database	C4.5	C4.5-L	C4.4	C4.4-X	C4.5-B	C4.4-B
Audiology	89.4±0.8	91.1±0.9	91.0±0.8	57.3±2.6	94.7±0.5	95.2±0.6
Breast	60.9±1.7	63.1±1.4	60.6±1.2	62.8±1.4	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	99.9±0.0
Credit	87.9±0.7	89.9±0.5	87.3±0.4	90.7±0.5	92.6±0.5	92.1±0.4
Diabetes	74.8±0.9	76.9±0.8	77.3±0.7	78.7±0.7	83.4±0.5	83.2±0.5
Echocardio	54.1±1.3	55.9±1.6	57.7±1.1	58.4±1.1	67.4±1.5	67.8±1.6
Glass	79.2±0.9	81.3±1.0	81.3±0.8	78.8±1.2	88.9±0.8	88.7±0.8
Heart	76.0±1.2	81.1±1.1	83.6±0.8	81.3±0.9	88.4±0.6	89.1±0.6
Hepatitis	64.3±2.5	68.4±2.2	76.7±1.5	71.7±1.9	83.2±1.4	84.0±1.4
Iris	96.0±0.6	96.9±0.3	97.3±0.4	97.2±0.4	99.0±0.2	99.2±0.2
LED	81.4±0.9	81.9±1.0	84.3±1.0	65.3±1.6	90.6±0.8	90.6±0.9
Liver	62.6±1.2	63.7±1.1	64.8±1.5	62.3±1.4	74.0±0.7	73.9±0.7
Lung	54.6±3.6	51.1±3.5	50.5±3.3	50.0±0.0	65.3±3.0	62.0±3.4
Lympho	79.7±1.4	83.0±1.5	84.7±0.8	82.8±1.2	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	100.0±0.0
Promoters	78.4±1.6	82.9±1.5	81.2±1.5	82.4±1.4	93.0±1.2	93.8±1.0
Solar	87.5±0.6	88.9±0.5	88.6±0.5	87.0±0.4	89.8±0.5	89.7±0.5
Sonar	70.5±1.3	76.2±1.4	76.5±1.4	75.2±1.7	85.2±1.4	84.5±1.3
Soybean	98.2±0.5	97.8±0.7	97.8±0.7	82.3±2.1	100.0±0.0	100.0±0.0
Splice	96.4±0.2	97.7±0.1	97.8±0.1	98.2±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	97.5±0.4	98.6±0.3
Tumor	68.8±0.7	71.7±0.7	68.5±0.8	63.1±1.0	77.0±0.7	76.0±0.6
Voting	97.1±0.4	98.2±0.2	94.6±0.7	97.9±0.2	98.6±0.2	98.9±0.1
Wine	94.3±0.6	94.5±0.7	94.4±0.8	94.3±0.8	99.4±0.1	99.4±0.1
Zoology	96.4±0.5	98.0±0.4	98.4±0.4	93.5±1.4	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
C4.4-X vs. C4.4	8 - 2 - 15	-3.3	5.0	3.0
C4.4-X vs. C4.5-L	9 - 1 - 15	-3.1	8.0	6.0
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

6.3 Pruning and Laplace correction

C4.4 is a very marked improvement over C4.5. Most of this improvement is due to the use of the Laplace correction, which, despite its simplicity, is extremely 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 (Bauer & Kohavi, 1999), who report that the use of an “ m -estimate Laplace correction” (Kohavi, Becker, & Sommerfield, 1997) reduces the mean-squared error (MSE) of PET probability estimates from 10.7% to 10.0%, averaged across fourteen data sets. The present results, using AUC, give a perspective complementary to those obtained with MSE. In addition, the uniformity of success of the simple Laplace correction (e.g., 21-2-2 for C4.5) is remarkable.

Not pruning outperforms pruning 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 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 KDD, the advantage of C4.4 will become stronger.

6.4 Chi-square pruning

However, as noted above, simply not pruning is not intuitively satisfying as the best method for training PETs. It seems that it would be more advantageous to modify the pruning to address the production of probability estimates directly.

We compared C4.4 with and without chi-square pruning, using the same data sets and methodology as above. The results were quite surprising. Chi-square pruning generally did not improve C4.4; more often, it degraded the probability estimates. This result holds across the entire spectrum of pruning thresholds (chi-square p values); we tried thresholds of 0.1%, 1%, 5%, 10% and 20%, with and without the Laplace correction (only the results for 5% with the Laplace correction are shown in Table 1). As with the comparison with C4.5, we believe these results may be due to the small size of the UCI data sets. C4.4 with chi-square tends to prune a lot, even with a high significance threshold like 20%, because after the first few levels there is not enough data for it to conclude with any reasonable confidence that parent and child distributions are significantly different.

We also compared C4.4 with a version of C4.4 that stops growing when the leaves become too small. Specifically, the C4.5 package provides a parameter m such that C4.5 will not split a node unless at least two of its children contain more than m (default 2) examples. A simple method for pruning is to increase m . Perhaps not surprisingly in light of the chi-square results, all values tried also underperformed C4.4.

6.5 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 mean-squared error (MSE) (Bauer & Kohavi, 1999). They show a decrease in the average MSE over fourteen data sets from 10.7% for regular PETs to 7.5% for bagged PETs. 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 work done by Provost, Fawcett and Kohavi (1998), who present the ROC curves of six algorithms evaluated on ten data sets. We observe that the ROC curves of bagged PETs (“bagged MC4”) have larger areas in their graphs. In fact, in all but one case, the bagged PETs completely dominate the curves of individual Laplace-corrected PETs (“MC4”).

It is noteworthy that the improvements in AUC with bagging are on average much larger

than the 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 the comprehensibility of the single tree is lost, and it 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.

7 Conclusions and discussion

The poor performance of PETs built by conventional decision-tree learning programs can be explained by a combination of two 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. Perhaps counter-intuitively (at first), larger trees can work better for probability estimation. We are disappointed that our results do not support the hypothesis that more accurate PETs can be built by using a pruning strategy designed specifically for improving probability estimation. We hope that future studies can explain this, perhaps by looking at larger data sets.

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.

These results have interesting connections to other recent work studying the relationship of model complexity and predictive performance. Oates and Jensen (1998) show that on UCI databases as the number of examples increases the accuracy of decision trees soon stabilizes, but decision-tree complexity (number of nodes) continues to increase. Our results present an important caveat: although larger trees may not be more accurate, that does not mean

that they are not better models. As shown by the results on the synthetic data, larger trees often model the problem much better even though they have equivalent accuracy.⁶ Apte et al. (1999) have also noted recently that when building rule-based and decision-tree-based probability estimators, the quality of the probability estimates continues to increase as more and more data are used for training—far beyond the points observed by Oates and Jensen, and in fact exhausting their 1.4 million data points without reaching a plateau.

Another significant observation is that bagged PETs produce excellent probability estimates. 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 0.1. We strongly echo the conclusion of Bauer and Kohavi (Bauer & Kohavi, 1999) that for problems where probability estimation is required, one should seriously consider using 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 validate the procedure used. As another example, the smoothing obtained by bagging the estimates, along with the increase in their accuracy, will help with probabilistic ranking (e.g., of interesting documents), for which the coarse estimates of small trees are particularly problematic.

8 Limitations, extensions and future work

The purpose of this work was to study how the probability estimates obtained by decision trees could be improved. We believe that the results we have presented have given us a substantially better understanding. However, what we have not yet studied is how these PETs compare with other methods for estimating probabilities. We hypothesize that as long as there are many examples, PETs can compete with more traditional methods for building

⁶This does not contradict the results of Oates and Jensen, who show that conventional decision-tree inducers build very large trees even from random data (Oates & Jensen, 1998).

class probability estimators, especially for high-dimensional problems (where decision trees typically excel, comparatively). We are especially interested in a comparison of bagged PETs with traditional methods. Their performance is particularly impressive in our study. However, it may just be that plain-old PETs still do not produce very good probability estimates. If this is the case, moving to methods for smoothing more sophisticated than the Laplace estimate may be worthwhile (Simonoff, 1998; Jelinek, 1997).

There are two possibilities that we have not yet tried that may improve the probability estimates of the bagged PETs even further. Breiman (Breiman, 1998) has noted that the estimates produced by bagged decision trees may be improved by using the 37% of the data held out of each bootstrap sample to obtain better estimates at the leaf nodes (because these data were not used for training). Also, more complex smoothing algorithms (such as averaging a leaf's estimates with those of its ancestors in the tree, with appropriate weights (Jelinek, 1997)) may do significantly better than the simple Laplace correction.

Finally, since we began by listing comprehensibility as one of the attractive features of decision trees, it is important to note that our strongest conclusion (bagged PETs work very well) involves an opaque combination of multiple trees. One method for producing a comprehensible model of a multiple-model classifier is to use it to label examples, and then learn from these new data (Craven, 1996; Domingos, 1997). For PETs the procedure would have to be modified slightly, since the learning task would be learning probabilities from probabilities. Of course, even C4.4-style PETs may be less than comprehensible, given their large size.

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.
- Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, *36*, 105–142.
- 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. <http://www.ics.uci.edu/~mlearn/-MLRepository.html>.

- Bradford, J., Kunz, C., Kohavi, R., Brunk, C., & Brodley, C. (1998). Pruning decision trees with misclassification costs. In *Proceedings of ECML-98*.
- 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. (1998). Out-of-bag estimation. Tech. rep. Unpublished manuscript.
- Breiman, L. (1996). Bagging predictors. *Machine Learning*, 24, 123–140.
- 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.
- Clark, P., & Boswell, R. (1991). Rule induction with CN2: Some recent improvements. In *Proceedings of the Sixth European Working Session on Learning*, pp. 151–163 Porto, Portugal. Springer.
- Craven, M. W. (1996). *Extracting Comprehensible Models from Trained Neural Networks*. Ph.D. thesis, University of Wisconsin – Madison. Technical Report No. 1326.
- Danyluk, A., & Provost, F. (2000). Telecommunications network diagnosis. In Kloesgen, W., & Zytkow, J. (Eds.), *Handbook of Knowledge Discovery and Data Mining*. To appear.
- Domingos, P. (1997). Why does bagging work? A Bayesian account and its implications. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pp. 155–158 Newport Beach, CA. AAAI Press.
- Domingos, P. (1999). MetaCost: a general method for making classifiers cost-sensitive. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 155–164.

- Domingos, P. (1997). Knowledge acquisition from examples via multiple models. In Fisher, D. H. (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. In *Proceedings of the Seventeenth International Conference on Machine Learning*, pp. 239–246. Stanford, CA: Morgan Kaufmann.
- Friedman, N., & Goldszmidt, M. (1996). Learning Bayesian networks with local structure. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, pp. 252–262. Portland, OR: Morgan Kaufmann.
- Fürnkranz, J. (1999). Separate-and-conquer rule learning. *Artificial Intelligence Review*, 13(1), 3–54.
- Hand, D. J. (1997). *Construction and Assessment of Classification Rules*. Chichester: John Wiley and Sons.
- 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.
- Heckerman, D., Chickering, M., Meek, C., Rounthwaite, R., & Kadie, C. (2000). Dependency networks for density estimation, collaborative filtering, and data visualization. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*. Stanford, CA: Morgan Kaufmann.
- Holte, R., Acker, L., & Porter, B. (1989). Concept learning and the problem of small disjuncts.. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 813–818. San Mateo, CA: Morgan Kaufmann.
- Jelinek, F. (1997). *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA.
- Jensen, D., & Schmill, M. (1997). Adjusting for multiple comparisons in decision tree pruning. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pp. 195–198.
- Kass, G. (1980). An exploratory technique for investigating large quantities of categorical data. *Applied Statistics*, 29, 119–127.

- Kohavi, R., Becker, B., & Sommerfield, D. (1997). Improving simple Bayes. In *The Ninth European Conference on Machine Learning*, pp. 78–87. Available: <http://robotics.stanford.edu/users/ronnyk>.
- 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.
- Niblett, T. (1987). Constructing decision trees in noisy domains. In *Proceedings of the Second European Working Session on Learning*, pp. 67–78 Bled, Yugoslavia. Sigma.
- Oates, T., & Jensen, D. (1998). Large data sets lead to overly complex models: an explanation and a solution. In Agrawal, R., & Stolorz, P. (Eds.), *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-99)*, pp. 294–298. Menlo Park, CA: AAAI Press.
- Pazzani, M., Merz, C., Murphy, P., Ali, K., Hume, T., & Brunk, C. (1994). Reducing misclassification costs. In *Proc. 11th International Conference on Machine Learning*, pp. 217–225. Morgan Kaufmann.
- Provost, F., & Fawcett, T. (1997). Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining (KDD-97)*, pp. 43–48. AAAI Press.
- Provost, F., & Fawcett, T. (1998). Robust classification systems for imprecise environments. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, pp. 706–713. Menlo Park, CA: AAAI Press.
- Provost, F., & Fawcett, T. (2000). Robust classification for imprecise environments. To appear in *Machine Learning*. <http://www.croftj.net/~fawcett/papers/ROCCH-MLJ.ps.gz>.
- Provost, F., Fawcett, T., & Kohavi, R. (1998). The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 445–453. Morgan Kaufmann. Available: <http://www.croftj.net/~fawcett/papers/ICML98-final.ps.gz>.

- 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. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.
- Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, California.
- Simonoff, J. (1998). Three sides of smoothing: categorical data smoothing, nonparametric regression, and density estimation. *International Statistical Review*, 66(2), 137–156.
- Smyth, P., Gray, A., & Fayyad, U. (1995). Retrofitting decision tree classifiers using kernel density estimation. In *Proceedings of the 12th International Conference on Machine Learning*, pp. 506–514.
- 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.moodysgra.com/research/crm/53853.asp>.
- Swets, J. (1988). Measuring the accuracy of diagnostic systems. *Science*, 240, 1285–1293.