

# Discovering Unrevealed Properties of Probability Estimation Trees: on Algorithm Selection and Performance Explanation

Kun Zhang<sup>1</sup>, Wei Fan<sup>2</sup>, Bill Buckles<sup>1</sup>, Xiaojing Yuan<sup>3</sup>, and Zujia Xu<sup>4</sup>

<sup>1</sup>EECS Department, Tulane University, New Orleans LA 70118, {zhangk, buckles}@eecs.tulane.edu,  
<sup>2</sup>IBM T.J. Watson Research, NY 10532, weifan@us.ibm.com, <sup>3</sup>College of Technology, Huston University,  
xyuan@central.uh.edu, <sup>4</sup>Computer Science Department, Dillard University, zxu@dillard.edu

## Abstract

*There has been increasing interest to design better probability estimation trees, or PETs, for ranking and probability estimation. Capable of generating class membership probabilities, PETs have been shown to be highly accurate and flexible for many difficult problems, such as cost-sensitive learning and matching skewed distributions. There are a large number of PET algorithms available, and about ten of them are well-known. This large number provides an advantage, but it also creates confusion in practice. One would ask “given a new dataset, which algorithm to choose and what performance to expect and not to expect? What are the reasons to explain either good or bad performance under different situations?” In this paper, we systematically, for the first time, answer these important questions by conducting a large-scale empirical comparison of five popular PETs by examining their AUC, MSE and error rate “learning curves” (instead of training-test split based cross-validation). Using the maximum AUC achieved by any of the evaluated probability estimation tree algorithms, we demonstrate that the preference of a probability estimation tree on different evaluation metrics can be accurately characterized by the “signal-noise separability” of the dataset, as well as some other observable statistics of the dataset explained further in the paper. Moreover, in order to understand their relative performance, many important and previously unrevealed properties of each PET’s mechanism and heuristics are analyzed and evaluated. Importantly, a practical guide for choosing the most appropriate PET algorithm given a new data mining problem is provided.*

## 1 Introduction

Inheriting the attractive properties of traditional decision tree induction, probability estimation trees or PETs can estimate the posterior probabilities  $P(y|x)$  instead of just a predicted class label. One of the simplest approaches to compute probability from a single decision tree is to divide the number of examples belonging to class  $y$  by the number of examples in the classifying leaf node. It is then up to the data-miners to make the optimal decision  $y^*$  which minimizes the expected loss or risk  $E_t[L(t, y^*)]$  based on the probability estimates and the predefined loss function  $L(t, y)$ . For many problems, the

true label  $t$  could be nondeterministic and different values of  $t$  may be given when  $x$  is sampled repeatedly. Typical examples of loss functions in data mining are 0-1 loss and cost-sensitive loss. For 0-1 loss function, the optimal decision is the most probable label or the label that appears the most often when  $x$  is sampled repeatedly. For cost-sensitive loss, the optimal prediction is the one that minimizes the empirical risk.

One important advantage of PETs, particularly as compared to parametric or generative methods, is that the hypothesis itself is much less “biased” in the sense that both the structure of the tree and the internal statistics are estimated from the labeled training data. However, parametric models usually assume that the true and typically unknown probability distribution follows a “particular form”. In other words, the formula is fixed and learning is to estimate the parameters of the formula using maximum likelihood estimation. In practice, given a new dataset with little knowledge about the true form of the target probability distribution, PET type of methods provide a family of rather unbiased, flexible and convenient solutions, and indeed their efficacy has been shown in various works, described below.

The utilities of PETs have been demonstrated in previous works to be highly accurate for many difficult problems. An incomplete list of these applications includes extremely skewed distribution [19], cost-sensitive learning [4,6,7], distributed learning [20] and concept-drifting data stream mining [21]. Many researchers have previously proposed a large number of PET algorithms, and around ten of them are relatively well-known. Although this large variety provides choice, it also imposes the inevitable confusion in practice. An important question that data miners have to confront with is: given a dataset, which algorithm to choose and what performance to expect and not to expect? Answers to this question are crucial to the successful deployment of PETs in practice.

In this paper, we systematically answer the above question by carrying out a large-scale empirical comparison of five popular PETs by examining their “learning curves” on AUC, MSE, and error rate. Those five algorithms include C4.5[1], C4.4[3], and confusion factor tree (CFT)[8], which are representative of single tree PETs, as well as random decision tree (RDT)[7] and

probability-averaged baggingPET (instead of majority voting) [3,15] that are often used as ensemble PETs. Each chosen algorithm is exhaustively examined on seventeen real-world datasets, varying in size, feature characteristic and noise level. Unlike some previous works that base algorithm comparison on a “fixed” training set size in either cross-validation or training-test splits, “learning curves” are the fundamental empirical basis for comparison used in this paper. To avoid overly simplified conclusion that “method A is always better than method B”, multiple performance metrics are being evaluated against each chosen PET. Quite different from cross-validation and training-test splits that use a “fixed” data size, learning curves represent the generalization performance of different models produced by the same learning algorithm as a function of the size of the training set. In other words, from each training set of a different size, a new model is constructed by the same algorithm. By examining the learning curves over a number of different datasets, the correlation between performance metric and training set size can be observed and possibly generalized over different datasets. Therefore, the conclusions will be unbiased on training set sizes.

Importantly, via the maximum AUC achieved by any of the evaluated probability estimation tree algorithms, we demonstrate that the preference of a probability estimation tree on different evaluation metrics, i.e., which algorithm is better, can be clearly characterized by the “signal-noise separability” of the dataset. The concept of signal-noise separability, originated from signal detection theory and discussed in more detail below, provides a good analogy for two different populations present in every learning domain with uncertainty – correct identification of information of interest and some other noise factors which may interfere this identification. Finally, a practical guide is suggested to data miners on how to choose the most appropriate PETs in light of the problem at hand. In summary, the basic procedure is to first approximate the signal-noise separability of the dataset by applying either RDT or baggingPET, without worrying about other algorithms for the moment. If the estimated signal-noise separability of the dataset is quite high, baggingPET is preferred for ensembles and C4.5 or C4.4 is a good choice for single trees. Otherwise when the separability is low, depending on the preference of ensembles or single trees, either RDT or CFT should be selected.

## 2 An Index for Signal-Noise Separability

In signal detection theory [18], correct identification of information of interest and random matches resulted from noise factors are represented by two different distributions. For example, imagine that a doctor is looking for evidence of a tumor by examining a CT image. Either there is a tumor (signal present) or not (signal absent). Based on the “yes” or “no” decision made by the doctor,

these two distributions are divided into four different regions (Figure 1a), hit, miss, correct reject, and false alarm. Hit ( $P(\text{“yes”}|\text{tumor})$ ) and correct rejection ( $P(\text{“no”}|\text{tumor absent})$ ) are the desired actions. Although relative areas of the four different outcomes vary as the decision criterion changes, the separation of the two distributions does not. By changing the decision criterion, the full range of the obtained hit and false alarm ( $P(\text{“yes”}|\text{tumor absent})$ ) rate can be visualized through the ROC (Receive Operating Characteristics) curve. Figure 1b shows two different ROC curves, each corresponding to a different signal strength affected by the different overlapping degree with noise demonstrated in Figure 1a. It is evident that the more signal overlaps with noise, the smaller the area under the ROC curve is and vice versa. As a consequence, the fraction of the total area under the ROC curve – AUC can serve as a concise index for the separability of signal from noise with respect to the domain under consideration. The larger the AUC score can be accomplished by a model, the more likely that the signal (class of interest) can be separated from the noise (class of no interest) within the underlying domain. As such, this domain is characterized as high degree of signal separability. In nature, a domain of high-degree (low-degree) signal separability can be either deterministic (stochastic) or of little or no noise (rather noisy), however, which factor determines the degree of separability is actually unknown. This is typically the case for problems in real-world applications.

Most studies in recent machine learning research simply focus on utilizing AUC score as a criterion for model comparison, while, to a great extent, the efficacy of AUC as the separability indicator of the underlying domain has been neglected. The only work that we are aware of which extensively employs this property of AUC is [14] and their main objective is to identify the preference between two different types of algorithms - logistic regression and tree induction. Discriminating the superiority among the same category algorithms is obviously more difficult, and to the best of our knowledge, this is the first time that AUC utilized as the separability quantifier for preference identification solely for the probability estimation tree algorithms. Therefore, within our context, the signal-noise separability of a dataset is indicated by the maximum AUC score achieved by any of the probability estimation trees involved in this study.

## 3 PET Algorithms

Given a target probability distribution  $P(y|x)$ , which is typically unknown, the inductive learner looks for the best model  $\theta$  within a hypothesis space to best approximate this true but unknown distribution. The estimated probability has none-trivial dependency on  $\theta$ , so the estimated probability is best denoted as  $P(y|x, \theta)$ . In our case,  $\theta$  is each PET. Obviously, the accuracy of the estimated probability is dependant on  $\theta$ .

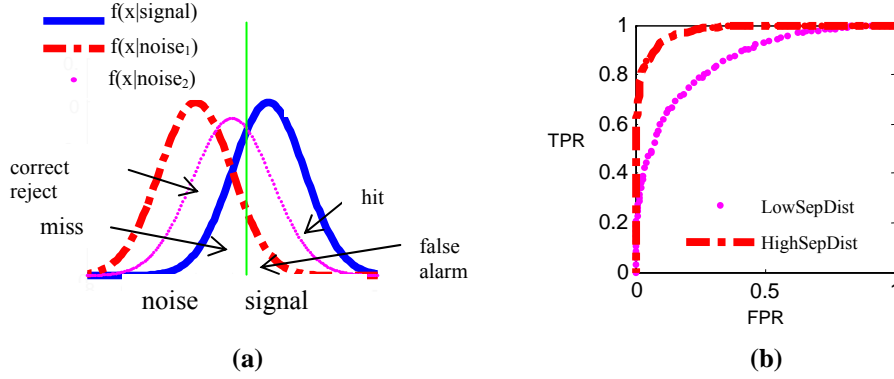


Figure 1

Typically, there are two types of probability estimation trees - a single tree estimator and an ensemble of multiple trees. The single tree is better for comprehensibility. However, it only approximates the true target function with one particular model representation and may not consistently achieve high accuracy. The ensemble is preferred when accurate probability estimation is desired. C4.5, C4.4 and CFT are representatives of single PETs. BaggingPETs and RDT are examples of ensembles.

### 3.1 C4.5 and C4.4

A straightforward method to estimate class membership probability from a decision tree is to use the frequency of the class at the classifying leaf node. If an unlabeled example  $x$  falls into a leaf node with  $M$  examples in total, of which,  $m$  examples belong to the positive class, then the posterior probability is simply  $P(+|x, \theta) = m/M$ . This method is widely used in popular decision tree systems, such as C4.5 and CART [2]. It has been noted [3] that frequency-based estimates of class-membership probability are not always accurate and statistically reliable. One reason is that the tree-growing algorithm searches for “pure” leaves, and tends to produce unrealistically high probability estimates, especially when the leaves cover just a few training examples. These estimates can be smoothed to mitigate these problems. As one of the simplest forms of smoothing, Laplace correction incorporates an equal prior for each class. Various experiments have showed that smoothing techniques can generally improve the performances [3,22]. Importantly, Provost et al [3] pointed out that many heuristics of decision tree induction used for improving classification accuracy and minimizing tree size actually “are biased against building accurate PETs”. Pruning via error reduction is blamed as the culprit. C4.4 [3] is essentially a variation of C4.5 by replacing frequency estimation with Laplace correction and turning off pruning and collapsing.

### 3.2 Confusion Factor Tree (CFT)

Aiming at improving the AUC value, Ling et al [8] propose the “confusion factor tree” (CFT) by assigning a fixed confusion factor  $s$  to each internal node of the tree.

This value measures the probability of errors altering the value of a tested attribute at a decision node due to noise introduced in data collection. Therefore, the probabilities from all other leaves as well as the probability of the leaf into which an example falls will contribute to the final probability estimate of the example. The contribution is determined by the number of unequal attribute values along the path leading to the leaf. Thus, the final class probability estimate for  $x$  is a weighted average of the contributions from all of the leaves in a tree:  $P(y|x, \theta) = (\sum \hat{P}_{L_i} \times s^q) / \sum s^q$ , where  $\hat{P}_{L_i}$  is the probability at leaf  $L_i$  and  $q$  is the number of unequal attribute values.

### 3.3 Random Decision Tree

Random decision tree [7] is an ensemble of individual trees, each determined randomly. To build each tree, a feature is randomly selected from remaining features with which to split data at a node. Along a decision path, a discrete feature can only be selected once; however, continuous features can be used multiple times, but each time with a different, randomly chosen splitting value. Features from the training data are used to construct tree structures and the data themselves to update the class probabilities that these trees represent. These probabilities are simply used to keep track the number of samples that are classified by each node. For a test point, each tree produces a class probability. Probabilities from all the trees in the ensemble are averaged to generate the overall class probability estimate. It is pointed out [4] that thirty trees is sufficient and fifty trees may be necessary when the distribution is skewed. On average, the depth of a random tree is about half of the number of the features, whereby distinct features are most likely to be selected to maximize diversity of the ensemble.

### 3.4 BaggingPETs

It is known that decision trees suffer from high variability. Bagging was introduced by Breiman [16] to address this problem, and has been shown to often work well in practice [15]. Given a training set  $z$  with  $n$  data points, bagging generates an ensemble by creating a number of bootstrap samples through uniform sampling with replacement from  $z$ . From each bootstrap sample of

size  $n$ , a tree is built. When a test point is evaluated, instead of the seminal bagging work that takes majority vote, bagging probability estimation trees uniformly average the probability of each class over all trees and the class with the highest probability is predicted for classification.

## 4 Evaluation Metrics and Approaches

Besides AUC, we have also considered the following metrics.

### 4.1 MSE

As an evaluation metric, AUC is essentially a measurement for probability “ranking.” It is not a proper assessment for the “accuracy” of probability estimation. MSE or Brier score is thus used to handle this problem. Given a data point  $x_i$ , MSE is calculated as  $1/N \sum_{i=1}^N \sum_y \{T(y|x_i) - P(y|x_i, \theta)\}^2$ , where  $T(y|x_i)$  is defined to be 1 if  $x_i$  of class  $y$ , and 0 otherwise. In addition,  $P(y|x_i, \theta)$  is the estimated probability for  $x_i$  to be a member of class  $y$ , and  $N$  is the size of a dataset.

### 4.2 Reliability Plots and Sharpness Histograms

By splitting the measurement space with respect to  $P(y|x, \theta)$ , MSE can be separated into two components – calibration and refinement [13]. Calibration measures the absolute precision of probability estimation, and refinement indicates how confident the estimator is in its estimates. Given two probability estimators of the same calibration capability, the one whose probability estimates are closer to 0 or 1 is more desirable. As a summary, calibration of the probability estimator can be visualized through “reliability plots”. “Binned” probability estimates are plotted horizontally versus the empirical probability calculated as the frequency of test points within the class of interest distributed in each bin. Empirical probability is a useful estimate for true probability, since true probability is not available for most real-world problems. If the estimator is well-calibrated, all points fall onto the diagonal line, indicating that the estimated probability equal to the empirical probability based on test samples. Refinement can be illustrated through “sharpness histograms” by plotting the relative frequency of the examples belonging to the target class versus the different levels of estimated probabilities denoted by the bins.

### 4.3 Error Rate

Error rate has been noticed to be an inappropriate criterion for evaluating probability estimates [11]. Error rate-related results are included so that this work can be related to past and future research that may only utilize this metric. For binary classification problem, error rate is usually obtained by thresholding the predicted probabilities at 0.5.

## 5 Experiment Design

We begin by describing the datasets used in this study

as well as the generation of learning curves and the implementation detail of each PET algorithm.

### 5.1 Dataset Description

To facilitate the calculation of AUC as well as reliability plots and sharpness histograms, we confine our study to binary classification problems. As shown in first column of Table 2, eighteen benchmark binary classification problems from UCI repository [10] are selected. These eighteen datasets, ranging from around one hundred to fifty thousands examples, include almost all natural occurring binary classification tasks in this database. From one dataset to another, different feature combinations and various marginal class distributions provide us a variety of domains to examine how the probability estimation tree models are affected by the characteristics of different datasets. Based on the feature property of each dataset, they can be divided into three groups: continuous, categorical and mixed. Continuous datasets refer to the datasets solely of numerical features, and categorical datasets are those which only include nominal attributes. Between them is the group of mixed datasets, whose feature vector includes both categorical attributes and numerical features. This categorization for each dataset is specified in the second column of Table 2. We treat the minority class of each dataset as the class of interest.

### 5.2 Learning Curve Generation

To generate rather smooth learning curves and fully utilize all available data points, for each dataset, we first reserve 25% random examples as the test set and use the remaining 75% data points as the pool for training sets. Each training set is then formed by random sampling from this pool, and each time the size of the training set increases by 10%. This procedure is repeated ten times for each dataset. Stratified sampling is utilized to ensure each obtained sample set conforms to the original class distribution. The same training and test sets are used for all of the PETs, and reported results are averages over the ten individual runs.

The learning curve of each algorithm is formed by connecting the corresponding mean of the ten-time runs for each training set size. For significance test, 95% confidence interval for the mean is used as a measure of the inherent variability of each evaluation metric across different training sets of the same size, and error bars are constructed at each training set size representing  $\pm t_{(\alpha/2, M-1)} \sigma / \sqrt{M}$ , where  $\sigma$  is sample standard deviation,  $M$  is sample size 10,  $t_{(\alpha/2, M-1)}$  is the upper critical value of the  $t$  distribution with  $M-1$  degrees of freedom. And,  $\alpha$  is set to be 0.05 to achieve the 95% confidence interval. During the evaluation process, two trees are considered to be different for a particular training-set size if the mean of neither falls within the error bars of the other.

### 5.3 Implementation Details of PET Algorithms

We implement all the PET algorithms described in Section 3. Our codes are based on C4.5 release 8. For C4.5, both the pruned (C4.5P) and unpruned (C4.5UP) trees with default settings are investigated. In addition, we construct FullC4.5 by disabling collapsing and pruning to make it more comparable with C4.4. Four variations of confusion factor tree are built for comparison, pruned (CFT.LC.P) and unpruned (CFT.LC.UP) CFTs with Laplace correction, as well as pruned (CFT.FE.P) and unpruned (CFT.FE.UP) CFTs with frequency estimation. As suggested in [8], we set the confusion factor to be 0.3. In addition to random decision tree with half depth (RDTH), we also implement RDT with full depth (RDTF). Bagging procedure is applied on C4.4 and FullC4.5 by turns, resulting in baggingC4.4 and baggingFullC4.5 consequently. Each ensemble includes thirty trees as base learners.

Before investigating the learning curves for these five PETs, we first conduct a preliminary study to identify which variant performs the best for different evaluation metrics as the training size increases. The top performer of each PET algorithm is then selected as the representative for the corresponding tree model. Due to space limits, we omit the results for this pilot study and Table 1 lists the best variant of each PET algorithm involved in the final study with respect to different evaluation metrics. Instead of the specific variants, generic names: C4.5, C4.4, CFT, RDT and baggingPET are used in the rest of the paper.

### 6 Experiment Results

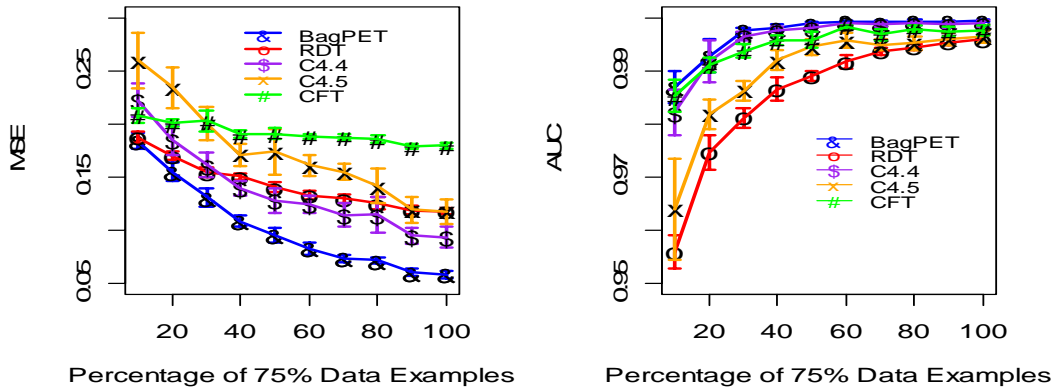
The main results for these eighteen datasets are summarized in Table 2, and each row corresponds to a dataset. The feature characteristic of each dataset is identified in the second column, and the third column lists the maximum AUC achieved by any of these five PETs. For each evaluation metric, the results from single PETs and ensembles are examined separately. This is because single PETs often subject to ensembles for almost all of the datasets. “Winner AUC” indicates the algorithm giving the best AUC for the largest training set, and for most of the training set sizes this algorithm dominates its counterparts. If two algorithms are listed, that means for most of the training set sizes, their results are statistically identical according to two-sided paired t-test at 95% confidence level. The same notations are applied to “Winner MSE” and “Winner ErrorRate”.

**Table 1:** Representatives for each PET algorithm involved in the study

PET Algorithms	Error Rate	AUC	MSE
C4.5	C4.5P	C4.5P	C4.5P
C4.4	C4.4	C4.4	C4.4
CFT	CFT.LC.P	CFT.LC.UP	CFT.FE.P
RDT	RDTF	RDTF	RDTF
BaggingPET	bagging FullC4.5	baggingC4.4	bagging FullC4.5

**Table 2:** Results of learning curve study for eighteen datasets

Dataset	Feature Types	MAX AUC	Winner AUC		Winner MSE		Winner ErrorRate	
			SinglePETs	Ensembles	SinglePETs	Ensembles	SinglePETs	Ensembles
Mushroom	Categorical	1	C4.4/C4.5	RDT/BaggingPET	C4.5	RDT/BaggingPET	C4.5/C4.4	RDT/BaggingPET
BC_wdbc	Continuous	0.995	CFT	RDT/BaggingPET	C4.4	RDT/BaggingPET	C4.5	RDT/BaggingPET
Chess	Categorical	0.99	C4.4/CFT	BaggingPET	C4.5/C4.4	BaggingPET	C4.5/C4.4	BaggingPET
BC_wisc	Continuous	0.99	CFT	RDT	C4.5/C4.4	RDT	C4.5/C4.4	RDT
HouseVote	Categorical	0.99	CFT/C4.4	RDT/BaggingPET	C4.5	BaggingPET	C4.5/C4.4	BaggingPET
Tic	Categorical	0.99	C4.4/CFT	BaggingPET	C4.4	BaggingPET	C4.5/C4.4	BaggingPET
Hypothyroid	Mixed	0.989	CFT/C4.4	RDT/BaggingPET	C4.5	BaggingPET	C4.5	BaggingPET
Spam	Continuous	0.98	CFT	RDT	C4.4	BaggingPET/RDT	C4.5	RDT
SickEuthyroid	Mixed	0.98	C4.4/CFT	BaggingPET	C4.5	BaggingPET	C4.5	BaggingPET
Ionosphere	Continuous	0.966	C4.4/CFT	RDT/BaggingPET	C4.4	BaggingPET	C4.5/C4.4	BaggingPET
Spectf	Continuous	0.96	CFT	RDT	C4.4	RDT/BaggingPET	C4.5/C4.4	RDT/BaggingPET
Australian	Mixed	0.934	CFT	RDT/BaggingPET	C4.5	BaggingPET	C4.5	BaggingPET
Adult	Mixed	0.9	CFT	RDT/BaggingPET	C4.5	BaggingPET	C4.5	BaggingPET
Sonar	Continuous	0.88	CFT/C4.4	RDT	CFT/C4.4	RDT/BaggingPET	CFT/C4.4	RDT
Hepatitis	Mixed	0.87	C4.4/CFT	RDT/BaggingPET	CFT	RDT/BaggingPET	C4.5/CFT	RDT
Pima	Continuous	0.825	CFT	RDT/BaggingPET	CFT/C4.4	RDT	CFT	RDT
Spect	Categorical	0.816	CFT	RDT	CFT	RDT	C4.5/CFT	RDT/BaggingPET
Liver	Continuous	0.75	CFT	RDT/BaggingPET	CFT	RDT	C4.5/CFT	RDT/BaggingPET



**Figure 2:** Plots illustrate that high signal separability datasets can be detrimental to the performances of CFT and RDT  
*Left:* MSE learning curves for dataset Tic; *Right:* AUC learning curves for dataset Chess.

Noticeably, if we choose 0.9 as the threshold to distinguish the datasets of high signal separability ( $\text{Max\_AUC} \geq 0.9$ ) from the low ones ( $\text{Max\_AUC} < 0.9$ ), the relative performances of these PETs seem to exhibit fundamental differences in these two groups. This is particularly true when feature set characteristics of each dataset are also taken into account. The datasets of low signal separability are indicated in italic font in Table 2. The Mushroom dataset is excluded from our discussion, since its max AUC is equal to 1, and all PETs quickly achieve the best statistically identical performances with respect to each evaluation metric.

### 6.1 Datasets of High Signal Separability

For the AUC metric, across the ten different training set sizes, baggingPET dominates RDT for AUC on categorical or mixed datasets - Chess, Tic and SickEuthyroid. On the other categorical or mixed sets, both algorithms achieve AUC scores which are statistically identical. On the categorical dataset with a small number of feature values - Chess, RDT even fails to the single trees as shown in Figure 2. While for continuous datasets - BreastCancer\_Wisc, Spam and Spectf, RDT outperforms baggingPET. Among single trees, CFT achieves overall better AUC scores and its strength especially exhibits on datasets of continuous or mixed features. C4.5 never wins its counterparts on a single dataset for AUC.

For the MSE metric, RDT is superior to baggingPET on BreastCancer\_Wisc. The datasets for which both algorithms perform statistically identically are Spectf, BreastCancer\_wdbc and Spam. These four sets are solely of continuous feature values. For continuous datasets, the only exception that baggingPET outdoes RDT is Ionosphere. On categorical or mixed datasets, baggingPET is consistently better than RDT. Among single trees, CFT never wins C4.5 and C4.4 on a single dataset of high signal separability. Its degraded performance of MSE on data set Tic is illustrated in Figure 2.

With respect to error rate, what we observed is quite similar to that for MSE. The datasets that RDT outperforms baggingPET or both perform statistically identically are BreastCancer\_Wisc, Spam, BreastCancer\_wdbc and Spectf. All of them are continuous datasets, and Ionosphere is again the sole exception. BaggingPET still exhibits advantages on categorical or mixed datasets. Among single trees, C4.5 is obviously the top performer, and CFT fails to C4.5 and C4.4 on all datasets of high signal separability.

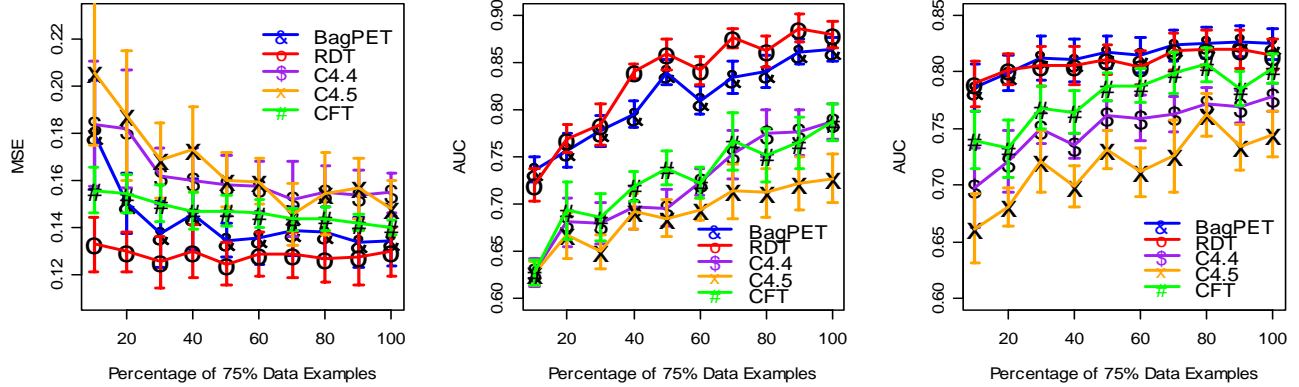
### 6.2 Datasets of Low Signal Separability

Compared to datasets of high signal separability, the phenomenon on low signal separability problems is “simpler”. On these five sets, baggingPET never outdoes RDT on any single dataset with respect to all three metrics. Their performances are at most statistically identical for some datasets. Similarly, among single trees, CFT shows clear advantages on these datasets, and it doesn’t lose to other single tree methods on any single dataset for all three evaluation metrics. Three learning curves of different datasets are provided in Figure 3 to illustrate the favorable performances of RDT or CFT on low signal separability datasets.

## 7 Conjectures and Analyses

By summarizing the above observations, we could derive the following conjectures:

1. In ensembles, RDT exhibits overall better performance on AUC. As show in the experiments, the averaged Win-Loss-Tie statistics of RDT versus baggingPET over the ten different training set sizes is 5.6-3.2-9.2
2. With respect to AUC, MSE and error rate, baggingPET is able to achieve better performance on categorical or mixed datasets of high signal separability. On the other hand, RDT shows advantages on datasets of low signal separability or continuous datasets of high signal separability.



**Figure 3:** *Left:* MSE learning curves for dataset Spect, illustrating a situation where RDT and CFT perform better on datasets of low signal separability; *Middle:* AUC learning curves for dataset Sonar, illustrating RDT dominates baggingPET for most of the training set sizes on a low signal separability dataset. *Right:* AUC learning curves for dataset Pima, illustrating CFT outdoes its counterparts on a dataset of low signal separability

- For datasets of high signal separability, RDT’s performance in terms of AUC, MSE and error rate can be affected adversely by the datasets solely made up of categorical feature values. This is especially true if each feature has a rather small number of values. In the experiment studies, both Chess and Tic datasets are two examples on which RDT performs poorly, and in both cases, each categorical feature takes from only two possible values.
- Among single trees, CFT achieves overall better AUC scores than C4.4 and C4.5. For MSE and error rate metrics, CFT achieves better results on datasets of low signal separability.

As follows, we explain these important conjectures by “digging” into the mechanisms employed by each PET algorithm. A thorough understanding can help us provide a useful heuristic to choose the most appropriate PET given a new dataset. Indeed, we provide such a heuristic in Section 8.

### 7.1 Why RDT and CFT Better on AUC?

Why does RDT outperform baggingPET and why does CFT perform the best among single trees for the AUC criteria? The explanation lies in the two factors which play the critical roles in the computation of AUC. As is well understood, ROC curve shows the ability of an algorithm to rank positive instances as compared to negative instances. The more positive examples are ranked higher than those negative examples, the higher the AUC score is, in general. For a random probability estimator, its ROC points would fall on the diagonal line  $x = y$ , whereby it has an AUC value around 0.5. All of the five PET algorithms we discussed are better probability estimators than a random guesser. None of them generates an AUC score less than or equal to 0.5 at any training set size for any datasets in our study. However, this is not enough. The probability estimates should vary from one test point to another, so that each example can

be ranked differently, and the number of unique probabilities is maximized as a result. This is quite important, since only unique pairs of true positive rate and false positive rate, or “plotted point” on ROC, resulting from thresholding by different probability estimates are considered in the trapezoidal integration rule for AUC calculation. Descriptions on AUC computation can be found in [12].

Based on the above analysis, we examined the number of unique probabilities that each PET can generate at different training set sizes with respect to each dataset. As a summary, Table 3 presents the average values and standard deviations of the pair-wised Win-Loss-Tie statistics over the ten different training set sizes. For 14.9 out of 18 sets, the number of unique probabilities generated by RDT is significantly larger than that of baggingPET. Moreover, baggingPET neither outdoes RDT on a single set nor even at a training set size. Quantitatively, the Win-Loss-Tie record of baggingPET versus RDT is 0-14.9-3.1 with a standard deviation of 0-0.9-0.9.

Utilizing distinct randomization methods at different stages of tree construction for ensemble building [5,6], RDT and baggingPET essentially can be unified by the framework that Breiman established for random forest [9]. In terms of the strength of the individual tree ( $S$ ) and the mean correlation among these trees ( $\bar{\rho}$ ), the generalization error of random forest is upper-bounded by  $PE^* \leq \bar{\rho}(1 - S^2)/S^2$ . This upper bound justifies that no ensemble can do better than the boundary given its strength and correlation. Lowering  $PE^*$  can be achieved by either minimizing  $\bar{\rho}$  or increasing  $S$ . In light of this framework, the reason that RDT can generate more unique probabilities than baggingPET is because, in general, the member trees of RDT are less correlated with each other than those of baggingPET. Each random

**Table 3:** Average values and standard deviations of the pairwise Win-Loss-Tie statistics by comparing the algorithms in the row versus the algorithms in the column over ten different training set sizes for unique probabilities.

Unique Probabilities				
AVG	RDT	C4.4	C4.5	CFT
BagPET	0-14.9-3.1	18-0-0	18-0-0	11.6-4-2.4
RDT		18-0-0	18-0-0	16-0.6-1.4
C4.4			17.9-0-0.1	0.1-15.2-2.7
C4.5				0-17.3-0.7
STDEV	RDT	C4.4	C4.5	CFT
BagPET	0-0.9-0.9	0-0-0	0-0-0	1.9-1.8-0.5
RDT		0-0-0	0-0-0	0.8-0.5-0.7
C4.4			0.3-0-0.3	0.3-1.6-1.3
C4.5				0-1.9-1.9

decision tree is used more as a data structure to group and summarize the training data rather than a clearly motivated hypothesis in traditional decision tree learning to maximize/minimize a splitting criterion. However, compared with RDT, individual trees of baggingPET still extensively use the splitting criterion at each step of tree expansion. The “randomization process” itself reflects the fact that the same dataset can be summarized and partitioned in many ways. Since randomization does not specify any particular ways and orders on how the training data should be summarized, it imposes a weaker “inductive or hypothesis bias” than splitting criterion-based tree expansion mechanisms. Statistically, each independently constructed decision tree is thus quite uncorrelated, as compared to the member trees of baggingPET.

From Table 3, CFT is apparently the top performer among single trees, followed by C4.4 and C4.5. The strength of CFT for distinctive probability generation mainly comes from its underlying aggregation technique. By aggregating the weighted probability estimates from all of the leaves in a tree, CFT is able to assign different estimated probabilities to different data points, even if these points are classified by the same leaf node. This capability increases the number of unique probabilities that CFT can generate, and the common observation that the number of unique probabilities generated by a single tree can not exceed the number of leaves in a tree obviously is not true for CFT.

### 7.2 Why RDT Better on Low-Signal Separability Datasets?

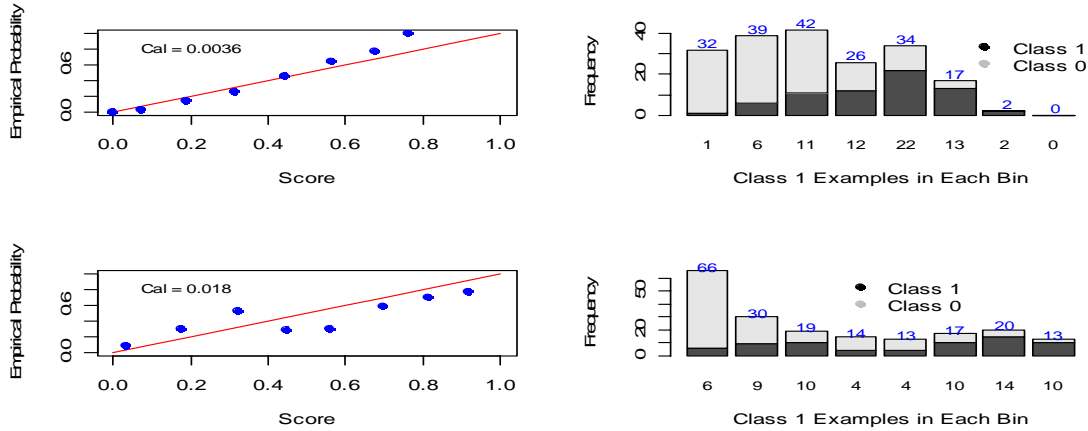
The efficacy that RDT exhibits on low signal separability datasets can be traced back to its underlying mechanism. Compared to the other four PETs, RDT is the only algorithm which forgoes any criterion for optimal feature selection at each internal node. Rather than “training” on the labeled examples as the other decision tree methods, RDT is more like a structure for data

“summarization.” When the separability between noise and signal is low, this property not only makes RDT avoid the massive searches or optimizations adopted by other PET algorithms, but also effectively prevents RDT from the danger of identifying noise as signal or overfitting on noise. In addition, according to the law of large number, RDT tends to provide an average of probability estimation which approaches the mean of true probabilistic values as more individual trees are added to the ensemble [5]. Thus, its estimated probabilities are less biased or less pure than those generated by baggingPET. As shown in Figure 4, when the problem is of low signal separability, RDT can clearly achieve better performance than other four PETs. On the other hand, if the domain is of high signal separability, and especially the domain involves some other characteristics which may adversely affect the diversity among individual random trees (explained in detail below), the performance of RDT, as illustrated in Figure 5, may be degraded and possibly inferior to that of baggingPET.

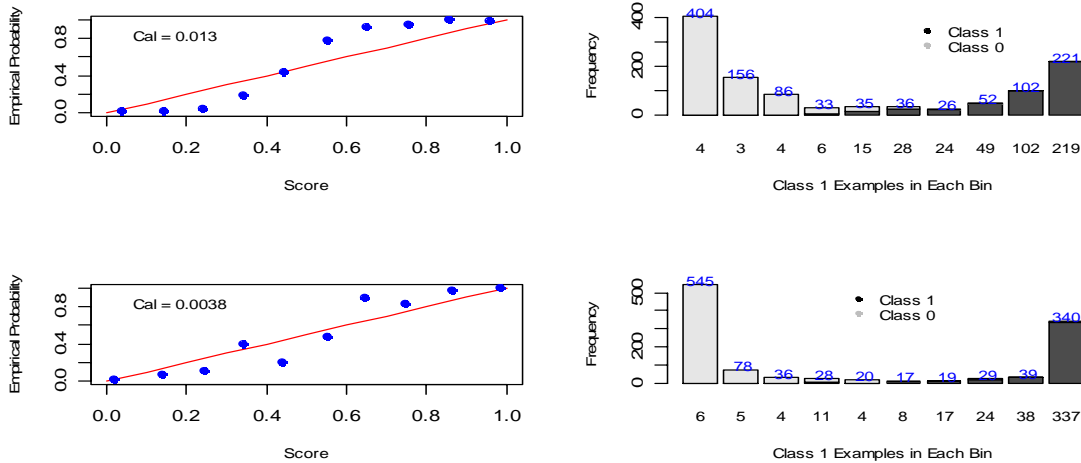
### 7.3 Why High-Signal Separability Categorical Datasets with Limited Feature Values Hurt RDT?

The reason that RDT’s performance may be detrimental on categorical sets with a small number of feature values can be ascribed to its inherent mechanism for random feature selection. Recall from Section 3.3, RDT is a tree ensemble which builds completely randomly. Each member tree is induced by completely ignoring any feature selection criterion that is employed by the other four PET algorithms. Along a decision path, a categorical feature can only be selected once; however, continuous features can be used multiple times, but each time with a different, randomly chosen splitting value. High separability datasets with a small number of categorical values tend to limit the degree of freedom or diversity that RDT’s random feature selection can explore. In other words, such datasets increase the correlation among each independently constructed single tree, and make these trees “more similar to each other”. As a consequence, the combination of these similar trees has restricted or no benefit on the reduction of variance, thereby the overall generalization error could be increased in terms of the upper bound derived by Breiman. Completely forgoing the criteria for optimal feature selection, RDT apparently aims at achieving higher tree diversity at the expenses of strength of each member tree, and the overall performance of RDT can be adversely affected if the diversity among member trees is impaired and strength of each tree is lost in the same time.

To further verify our conjecture, we discretize each feature of BreastCancer\_Wisc dataset into 10, 5 and 2 bins. The AUC learning curves of baggingPET and RDT for the original set of continuous features as well as these three discretized categorical feature sets are shown in



**Figure 4:** Plots illustrate the overall better performance of RDT on dataset of low signal separability indicated by smaller calibration error; **Upper:** Reliability plot and sharpness histogram of RDT on dataset Pima at 100% training set size; **Lower:** Reliability plot and sharpness histogram of baggingPET on dataset Pima at 100% training set size. (Class 1 represents the target class of interest. The number at the top of each bin indicates how many examples within this bin)

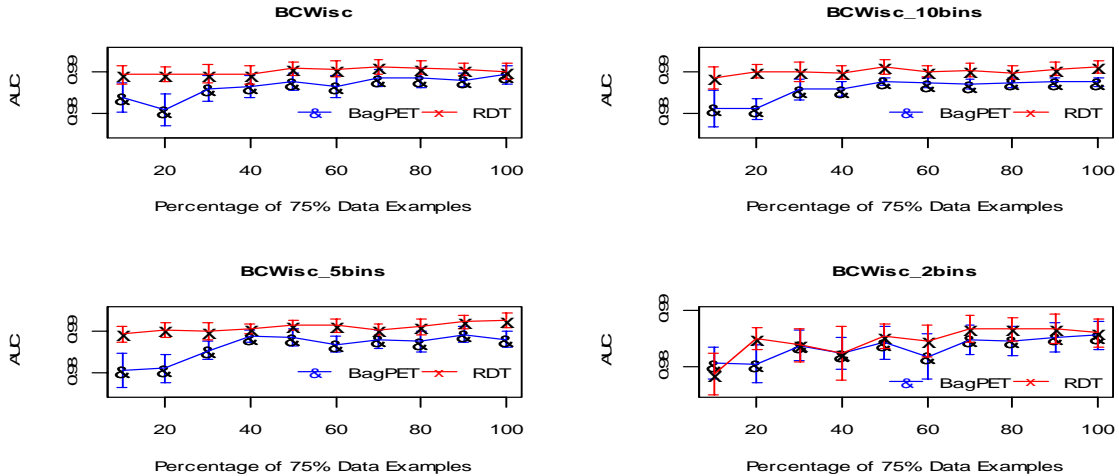


**Figure 5:** Plots illustrate the overall better performance of baggingPET on dataset of high signal separability indicated by smaller calibration error; **Upper:** Reliability plot and sharpness histogram of RDT on dataset Spam at 100% training set size; **Lower:** Reliability plot and sharpness histogram of baggingPET on dataset Spam at 100% training set size

Figure 6. Obviously, as compared to the AUC learning curves for the original continuous feature set, the changes resulting from discretization into 10 bins and 5 bins are trivial, which can be clearly seen in the respective plots. However, for both algorithms, the differences between the original continuous feature set and 2-bin categorical feature set are visually apparent and statistically significant. On the original continuous feature set, RDT achieves significantly higher AUC score than baggingPET across different training set sizes. However, their performances are statistically identical on 2-bin categorical feature set. RDT has an average AUC reduction rate 5.7% which is higher than the 3.35% reduction of baggingPET. This result provides the experimental evidence that RDT favors continuous datasets.

#### 7.4 Why CFT Favorable on Low-Signal Separability Sets?

The reason that CFT performs better on datasets of low signal separability is mainly due to the motivation and rationale of this algorithm. As stated in Section 3.2, by aggregating the estimated probabilities from all of the leaf nodes in a tree, CFT assumes that noise or error embroiled during data collection process may alter the attribute values, and the data points might have fallen into other leaves. Therefore, the estimated probability for each example should take this factor into account, and the contribution from each of the other leaves is reversely determined by the deviations between attribute values along the decision path leading to the classifying leaf node, and those along the path to the other leaf nodes.



**Figure 6:** The AUC learning curves of baggingPET and RDT for original BreastCancer\_Wisc set of continuous features and discretized categorical sets in which each feature has 10, 5 or 2 different values. This example empirically illustrates that RDT’s performance can be adversely affected by the categorical high signal separability datasets with a small number of values.

Although the low signal separability datasets in nature can be either noisy or stochastic, this inherent ensemble-like probability aggregation mechanism undoubtedly can help rectify the error introduced to the probability estimates simply due to the attribute noise, whereby CFT demonstrates good performances on this sort of datasets with respect to the three evaluation metrics. Figure 7 presents such a situation in which CFT is superior to C4.4. However, if the underlying domain is of low noise or really deterministic, aggregation of the estimated probabilities from the other irrelevant leaves will introduce undesirable error or noise into the final probability estimates, and this is typically the case when a more complicated tree model with a huge number of leaves is induced as the training set size increases. As more data points are assimilated, these errors tend to accumulate since the contributions from the other non-target leaves may account for a larger proportion while the target leaf contributes relatively less, thereby its overall probability estimation accuracy is affected reversely.

## 8 Heuristics on how to Choose PET for a New Problem

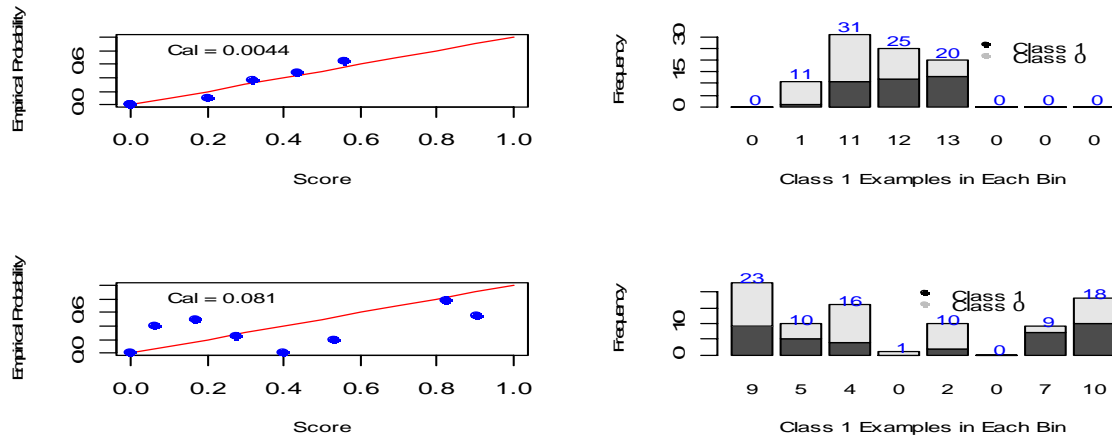
The discoveries in this paper obviously can help data mining practitioners choose the most appropriate PET algorithm given a new problem. The procedure is summarized in Figure 8. Specifically, the choice of which PET to use depends on the signal-noise separability of the given problem, practical preference for either single trees or ensembles, predefined evaluation metrics, as well as feature vector characteristics. By taking these factors into account, the reason for each algorithm’s preference is demonstrated in Section 6, and later further explained and analyzed in Section 7. For example, the superiority of RDT on low signal separability domains is first shown in Section 6.2 and then analyzed in Section 7.2.

As a brief “walk-through” of the procedure, the first step is to estimate the signal-noise separability of the given problem using all available training points. When the signal-noise separability is high, baggingPET is expected to achieve overall better performances on datasets with categorical feature, and RDT is preferred for datasets with continuous features. Among single trees, CFT is preferred over other methods if the criterion is to maximize AUC. While both C4.5 and C4.4 are better choices if either MSE or error rate is to be minimized. For low separability problems, either RDT or CFT should be selected regardless of the criterion.

This straightforward heuristic is based on extensive empirical studies and thorough performance analyses presented in previous sections. It is applicable to sixteen out of the seventeen sets used in this paper, captures the common patterns over a variety of problems, and is independent from training data size. On the utility side, this rule can save the trouble of exhaustive tests to choose the most appropriate PET algorithm for a new data mining problem.

## 9 Related Works

The concept of maximum AUC is first introduced in [17] to compare various learning methods on a medical dataset. Recently, Perlich et al [14] applied this criterion in a learning curve study which focuses on identifying the preference of logistic regression and decision trees according to the characteristics of data. Their results show that logistic regression performs better on datasets of low signal separability, while the high signal-separability situation is in favor of tree models. Independent of the marginal probability of class membership, maximum AUC is claimed to be a more robust criterion for Bayes rate estimation [14], and its magnitude, as we illustrated previously, can serve as a quantifier indicating the signal-noise separability of the underlying domain. Although in



**Figure 7:** Plots illustrate the preference of CFT on dataset of low signal separability indicated by smaller calibration error. *Upper:* Reliability plot and sharpness histogram of CFT on dataset Liver at 100% training set size; *Lower:* Reliability plot and sharpness histogram of C4.4 on dataset Liver at 100% training set size

general, we can not determine the actual nature of separability of a domain, maximum AUC indeed provides some clues of such inherent characteristics in a sensible way. Before we used this criterion in our study, we didn’t expect that even the same category models can be potentially distinguished according to this criterion, and a nature way to expand our study and the work of Perlich’s would be: on the datasets of low signal separability, which one is better? logistic regression, RDT or CFT? In [5], error-rate based evaluation of RDT, bagging, single tree and random forest is conducted on cross-validation based experiment. However, this paper utilizes learning curve (that is training set size independent), uses the idea of signal-noise ratio, evaluates error rate, AUC, MSE, reliability plots, etc, and also digs into the details of a rather exhaustive combination of situations to explain what works and why it works. In both [4] and [6], their main focus is on how PET could approximate the true probability, and experiments are mainly conducted on train-test splits. In [23], although several PETs are evaluated through learning-curves, they have different goals from us, and relatively limited explanation is provided on the reasons behind the results. Most recently, Davidson and Fan [24] showed that traditional bagging that uses majority voting doesn’t perform well in a number of situations, such as class label noise, multi-label, sample selection bias, and small training sets. It is interesting to study the performance of averaged probability version or baggingPET under similar situations.

## 10 Conclusions

In this paper, we have systematically addressed and answered the important questions that: “without an exhaustive test, which PET would be the most appropriate algorithm for a new problem? Why a PET perform well in one situation but not so well in another?” Answers not only help algorithm selection, but also explain the reasons

of either good or bad performance, thus serving as a guide to design better algorithms.

First, using maximum AUC score achievable by the evaluated probability estimation tree algorithms, we demonstrate that the preference of a probability estimation tree on several widely-used evaluation metrics can be effectively determined by the “signal-noise separability” of the dataset. Based on this categorization, some of the important discoveries on PETs relative performance are as follows. Between baggingPET and RDT, 1) baggingPET is expected to perform better on AUC, MSE and error rate if the dataset is either of only categorical feature values, or of both continuous and categorical values but the problem has high signal-noise separability, 2) while RDT is preferred on either datasets of low signal separability regardless of feature vector type, or high-signal separability datasets with continuous features. Among single trees, 3) CFT is expected to achieve higher AUC scores than both C4.4 and C4.5 for most problems, 4) For MSE and error rate, CFT is preferred over C4.4 and C4.5 only on datasets of low signal separability.

In order to understand the relative performance and generalize beyond these discoveries, systematic analyses have been provided to explain the suitability of the mechanism and heuristic employed by each algorithm under different situations (i.e., high or low signal separability, continuous or categorical feature set, etc). We have found that, by discarding any feature selection criterion, but gaining diversity through complete random feature selection, RDT is expected to perform well when the signal separability is low. On the contrary, by selecting features through information gain, and obtaining diversity from random manipulation of the training set, baggingPET is expected to work better than RDT when the signal-noise separability is high, so that it can benefit from the use of information gain. For single trees, CFT’s

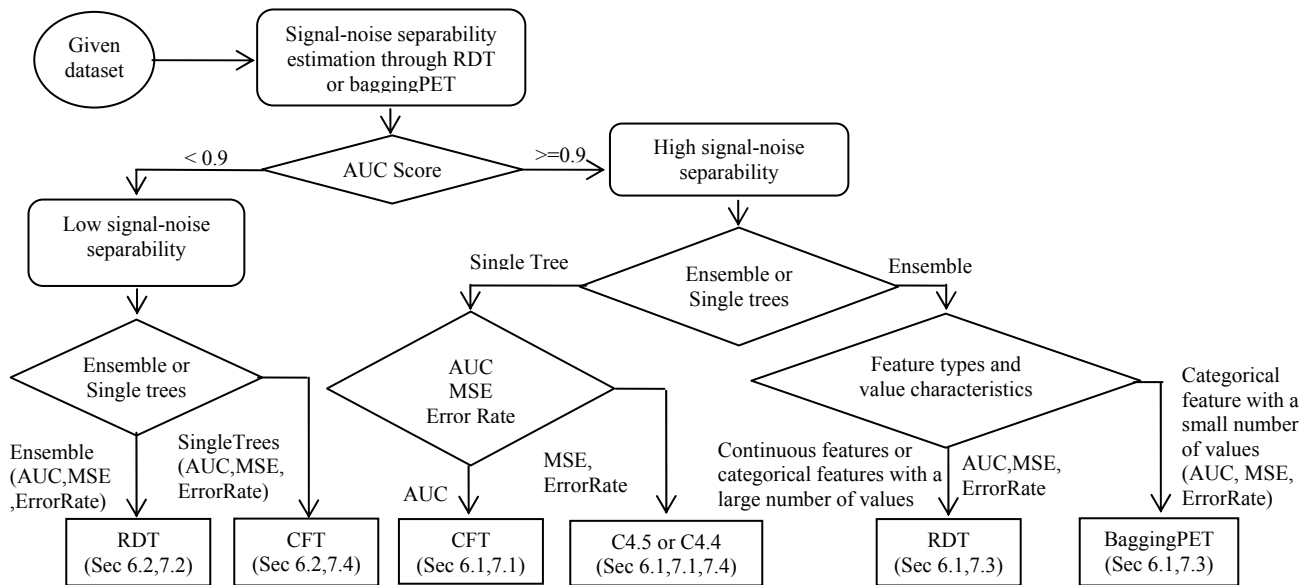


Figure 8: Procedure diagram for PET selection given a dataset

advantage on datasets of low signal separability mainly comes from its unique aggregation mechanism that does not rely on a single leaf node. However, for datasets of high signal separability, its performance is expected to be inferior to C4.4 or C4.5 due to this same aggregation. In addition, we have also shown that the performance of RDT can be sensitive to feature types, either continuous or categorical. The main reason lies in the diversity creation method incorporated within RDT that limits the diversity of trees built from purely categorical features. Finally, based on the extensive empirical results and analyses using seventeen datasets, a practical guide is provided to data mining practitioners to choose the appropriate PET algorithm given a new dataset without running into the trouble of exhaustive tests.

## References:

- [1] J. Quinlan. C4.5: programs for empirical learning. Morgan Kaufmann, 1993
- [2] L. Breiman., J. Friedman, R. Olshen., and C. Stone, Classification and Regression Trees. Chapman & Hall, 1984
- [3] F. Provost and P. Domingos. "Tree induction for probability based rankings", Machine Learning, 52(3), 2003, pp. 199-215
- [4] W. Fan, "On the optimality of probability estimation by random decision trees", AAAI 2004, pp.336-341
- [5] F. Liu, Master thesis, "The utility of randomness in decision tree ensembles", Monash University, 2005
- [6] W. Fan, E. Greengrass, J. McCloskey, P. Yu and K. Drummey, "Effective estimation of posterior probabilities: explaining the accuracy of random decision tree approaches", ICDM2005, pp.154-161
- [7] W. Fan, H. X. Wang, P. S. Yu and S. Ma, "Is random model better? On its accuracy and efficiency", ICDM 2003, pp.51-58
- [8] C. Ling and J. Yan, "Decision Tree with better ranking", ICML, 2003
- [9] L. Breiman. "Random forests", Machine Learning, 45(1), pp. 5 - 32, 2001.
- [10] C. Blake and C. Merz, "UCI repository of machine learning database", UCI, 1998
- [11] D. Hand, Construction and assessment of classification rules. Wiley, 1997
- [12] T. Fawcett, "ROC graphs: notes and practical considerations for data mining researchers". HPL-2003-4.
- [13] M. DeGroot and S. Fienberg, "Assessing probability assessors: calibration and refinement", Statistica Decision Theory and Related Topics III, Vol 1, 1982, pp. 291-314
- [14] C. Perlich, F. Provost and J. Simonoff, "Tree Induction vs. Logistic Regression: A Learning-curve Analysis", The Journal of Machine Learning Research, Vol. 4, pp 211, 2003
- [15] E. Bauer and R. Kohavi. "An empirical comparison of voting classification algorithms: bagging, boosting, and variants", Machine Learning, 36(1-2), 1999, pp. 105-139
- [16] L. Breiman, "Bagging predictors", Machine Learning, 24(2), pp 123-140, 1996
- [17] M. Ennis, G. Hinton, D. Naylor, M. Revow and R. Tibshirani, "A comparison of statistical learning methods on the Gusto database", Statistics in medicine, 17, pp: 2501-2508,1998
- [18] D. Green and J. Swets, Signal detection theory and psychophysics, Wiley, 1966
- [19] W. Fan, H. X. Wang, P. Yu, S. J. Stolfo, "A fully distributed framework for cost-sensitive data mining", ICDCS 2002, pp 445-446
- [20] W. Fan, P. S. Yu, H. X. Wang, "Mining extremely skewed trading anomalies", EDBT 2004, pp: 801-810
- [21] W. Fan, "Systematic data selection to mine concept-drifting data streams", KDD 2004, pp 128-137
- [22] C. Ferri, P.A. Flach and J.Orallo, "Improving the AUC of probabilistic estimation trees", ECML, 2003
- [23] K. Zhang, Z Xu, J. Peng and B. Buckles, "Learning through changes: an empirical study of dynamic behaviors of probability estimation trees". ICDM2005, pp. 817-820
- [24] I. Davidson and W. Fan, "When Efficient Model Averaging Out-Performs Boosting and Bagging", ECML/PKDD 2006