Effective Estimation of Posterior Probabilities: Explaining the Accuracy of Randomized Decision Tree Approaches

Wei Fan[1] Ed Greengrass[2] Joe McCloskey[2] Philip S. Yu[1] Kevin Drummey[2]

[1] IBM T.J.Watson Research, Hawthorne, NY 10532
{weifan,psyu}@us.ibm.com

[2] US Department of Defense, Ft. Meade, MD 20755
{edgreen,jpmccolo,drummey}@afterlife.ncsc.mil

Abstract

There has been increasing number of independently proposed randomization methods in different stages of decision tree construction to build multiple trees. Randomized decision tree methods have been reported to be significantly more accurate than widely-accepted single decision trees, although the training procedure of some methods incorporates a surprisingly random factor and therefore opposes the generally accepted idea of employing gain functions to choose optimum features at each node and compute a single tree that fits the data. One important question that is not well understood yet is the reason behind the high accuracy. We provide an insight based on posterior probability estimations. We first establish the relationship between effective posterior probability estimation and effective loss reduction. We argue that randomized decision tree methods effectively approximate the true probability distribution using the decision tree hypothesis space. We conduct experiments using both synthetic and real-world datasets under both 0-1 and cost-sensitive loss functions.

1 Introduction

Quite different from widely accepted single decision tree algorithms, the family of randomized decision tree methods introduces different methods of "randomization" into the decision tree construction process, and computes multiple decision trees instead of a single decision tree. A rather complete survey and comparison of these approaches can be found in [13]. Randomization has been explored in both data selection and model induction. Data randomization techniques include bootstrapping [4], feature subset randomization [1], data partitioning, and output perturbation [5]. In model induction, randomization has been explored in both total random feature selection as employed in Random Decision Trees (RD) [11], and partial random feature selection in Random Forest [6]. They are many publications and independent studies to support the finding that randomized decision tree methods are highly accurate. However, the missing piece is to understand the reason why randomized decision tree methods produce highly accurate models. Our work in this paper explores the reason behind the results by studying the relative effectiveness of "posterior probability estimation" by traditional decision tree and randomized decision tree ensembles. We believe our findings not only help us understand the behavior and limitation of randomized decision tree methods but also provide some insights into how to design more accurate algorithms.

2 Probabilistic View of Decision Trees

The basic procedure of decision trees and many related significant works can be found in [14]. We interpret decision trees as probability estimators and establish connections between probability estimation and loss minimization. A feature vector $\mathbf{x}$ has probability $P(y|\mathbf{x})$ to be a member of class y. By definition, $P(y|\mathbf{x})$ is the percentage of times that $\mathbf{x}$ has class label y when $\mathbf{x}$ is sampled exhaustively; $P(y|\mathbf{x})$ is determined by a typically unknown target function of the dataset (or the function that generates the true label for each example) and is independent of any modeling techniques and training examples. A decision tree (denoted by θ) can be regarded as an estimator to the true probability. The estimated probability depends on both the feature vector $\mathbf{x}$ and the decision tree θ, denoted by $P(y|\mathbf{x},\theta)$. Normally, $P(y|\mathbf{x},\theta) \neq P(y|\mathbf{x})$ unless the true target model is the decision tree θ itself or verified exhaustively for every $\mathbf{x}$. Assume that n is the number of examples at a leaf node classifying $\mathbf{x}$ and n_y is the number of examples among n with class label y, the estimated probability for $\mathbf{x}$ to be a member of class y is

$$P(y|\mathbf{x},\theta) = \frac{n_y}{n} \quad (1)$$

3 Probability, Loss, and Optimal Prediction

Given a loss function $L(t,y)$ where t is the true label and y is the predicted label, an optimal decision tree is one that minimizes the average loss $L(t,y)$ for all examples, weighted by their probability. The optimal decision y_* for $\mathbf{x}$ is the label that minimizes the expected loss $E_t(L(t,y_*))$ for a given example $\mathbf{x}$ when $\mathbf{x}$ is sampled repeatedly and different t's may be given. If the true probability distribution $P(y|\mathbf{x})$ is given, we can use it to choose y_*. It implies that if the estimated probability is equal to the true probability for every example, i.e., $\forall \mathbf{x}, P(y|\mathbf{x},\theta) = P(y|\mathbf{x})$, the

prediction by θ will always be the optimal decision for every example under any loss function, and no other models would have lower expected loss.

In order to evaluate θ's performance to minimize loss functions, we shall compare the difference between $P(y|\mathbf{x})$ and $P(y|\mathbf{x},\theta)$. Clearly, if $P(y|\mathbf{x})$ is given, we just directly use their difference. However, in reality, class labels are provided in inductive learning, but true probability $P(y|\mathbf{x})$ is not given for most problems, and it is strongly biased to assume $P(y|\mathbf{x}) = 1$ from a single observation that $\mathbf{x}$ has class label y. Nonetheless, for a fixed loss function, exact values of $P(y|\mathbf{x})$ may not be necessary to predict the optimal decision y_*. As long as the estimated probability $P(y_*|\mathbf{x},\theta)$ is above "some threshold", the optimal decision y_* will be predicted. For two-class problems evaluated under 0-1 loss, y_* is predicted as long as $P(y_*|\mathbf{x},\theta) > 0.5$. Similarly, for a cost-sensitive credit card fraud detection problem, assume that the amount of transaction of $\mathbf{x}$ is \$1000, and the cost to challenge a fraud is \$90. The decision threshold to predict fraud is $P(\text{fraud}|\mathbf{x},\theta) \cdot 1000 > 90$ or $P(\text{fraud}|\mathbf{x},\theta) > 0.09$. Obviously, the threshold is dependent on both the loss function and example.

Definition 1 *Given an example* $\mathbf{x}$*, the decision threshold* $v(y)$ *is the minimal probability to predict class label* y*, i.e.,* $P(y|\mathbf{x},\theta) \geq v(y)$.

Since we are only interested in predicting the optimal decision y_* to minimize loss functions, we focus on the relative difference between $P(y_*|\mathbf{x},\theta)$ and $v(y_*)$. When $P(y_*|\mathbf{x},\theta) < v(y_*)$, the difference $|P(y_*|\mathbf{x},\theta) - v(y_*)|$ measures how far off one is in making the optimal prediction. On the other hand, when $P(y_*|\mathbf{x},\theta) \geq v(y_*)$, this difference is insignificant in the sense of reducing the given loss function.

One practical difficulty is that y_* is dependent on $P(y|\mathbf{x})$ and is not known either. However, we can assume y_* to be the true label t of $\mathbf{x}$ since predicting the correct class label is expected to reduce any reasonable loss function. It is important to understand that assuming $y_* = t$ is different from, and less biased than $P(t|\mathbf{x}) = 1$. In fact, $y_* = t$ implies that the true probability $P(t|\mathbf{x})$ is within $[v(t], 1]$ that encompasses $P(t|\mathbf{x}) = 1$. For example, assuming $y_* = t$ under 0-1 loss for a two-class problem is the same as assuming $P(t|\mathbf{x}) \in [0.5, 1]$. Another example showing that $P(t|\mathbf{x}) = 1$ is a strong assumption can be found in Section 5 paragraph 2.

4 Randomized Decision Tree Approaches

In this paper, we consider four different approaches of randomized decision trees and their variations. The spirits of many other approaches (reviewed in Section 7) are incorporated into these methods. One of the most "random" methods is **Random Decision Tree** [11] or **RD**. RD computes an ensemble of typically 10 to 30 decision trees. During tree construction, the splitting feature at each node is chosen randomly from any "remaining" features without calling any information gain or other gain function. A chosen discrete feature on a particular decision path (starting from the root of the tree to the current node) cannot be chosen again since it is useless to test the same discrete value more than once, i.e., each split path will have the same discrete feature value. However, continuous features can be chosen multiple times, each time with a different randomly chosen threshold value. In order to prevent the tree from growing unnecessarily large, the tree stops growing when either a node becomes empty or the depth exceeds a predefined limit (e.g., the total number of features). One example to show the advantage of expanding the tree without testing gain functions can be found in [11]. It is an XOR-type problem such that each feature has no distinctive value by itself, but they only exhibit information gain when examined collectively. When classifying an example, each tree outputs posterior probability as defined in Eq 1, then the probabilities from each tree in the ensemble are averaged as the final posterior probability estimate. The simplest implementation of random decision tree is to remove the information gain check and choose an available feature (none-chosen discrete feature or continuous feature with random split value) randomly. Its name may superficially suggest total randomness, but the randomness of RD is only on the structure of each tree, and each tree is still consistent with the training data.

When bagging is applied to decision tree (we call this **BT**), each tree is computed from a bootstrap sample of the training set. In the original proposal [4], each tree's classification counts towards a vote on the predicted class label. The one with the most number of votes will be the final prediction. Although the votes into different classes can be converted into probability by normalization, we use a probabilistic variation similar to RD, that is, each tree outputs a probability of each class and then the multiple probability outputs are averaged. We call the probabilistic variation of bagging **BT+**.

Random Forest or **RF** [6] introduces randomness into decision tree in both data randomization and feature selection. RF computes multiple trees, and each tree is constructed from a bootstrap sample and by selecting at each node the feature with highest information gain among k randomly chosen features at the given node from the total feature set. The parameter k is provided by the user. RF performs either simple or weighted voting on the final prediction. As we did with bagging, we also use a probablistic variation of Random Forest called **RF+** that sums posterior probabilities instead of adding votes.

The last chosen algorithm is **Disjoint Sample Trees** or **DST**. The training set is randomly partitioned into several disjoint subsets of equal size. A tree is constructed from

each partition. As for the other three methods, averaged probability is output as classfication. Different but signficantly similar forms of DST have been proposed and used in many situations.

Bayesian Interpretation As compared with a single tree, Randomized decision tree methods estimate the true probability using a subset Θ of decision trees consistent with the training data. Formally, randomized decision tree approaches compute the following

$$P(y|\mathbf{x},\Theta) = \sum_{\theta\in\Theta} P(y|\mathbf{x},\theta)\cdot P(\theta|D) = \frac{1}{|\Theta|}\sum_{\theta\in\Theta}\frac{n_{y,\theta}}{n_\theta} \quad (2)$$

Θ is the ensemble or subset of decision trees chosen by a given randomized decision tree algorithm. $P(\theta|D)$ is the probability of decision tree θ after observing training data D. Since no preference is given to any tree, each of them receives uniform posterior, $1/|\Theta|$. Eq 2 is similar to decision tree averaging [7], and is a specific form of Bayesian optimal classifier (BOC) [14] or Bayesian model averaging [12] applied to decision trees with a "smaller" hypothesis space. Though widely used by statisticians, BOC receives much less interests among data mining researchers, since it is deemed computationally prohibitive to enumerate every model. This is particularly true for decision trees. Assume a trivial dataset with only two features and each feature is bi-valued. The total number of unique trees that are consistent with the training set is as many as 8. When any feature is continuous, the number of trees consistent with a training set can be infinite. The four randomized decision tree algorithms obviously chooses a much smaller hypothesis space to avoid the computationally prohibitive task to enumerate every decision tree.

The discussion of the optimality of BOC (relative to single model) to estimate posterior probability can be found in both [14] and [12]. Even if the complete hypothesis space can be enumerated by an exhaustive BOC, there is still no guarantee that it will exactly produce $P(y|\mathbf{x})$ for every $\mathbf{x}$. For some problems, the true model that generates the labels may not be contained in the hypothesis space of decision trees. In this paper, we are interested in justifying that the specific choices of hypothesis space of randomized decision tree approaches approximate the true probability **better** than single decision trees. If it is indeed true, the findings provide insights to find a possibly better subset of trees than existing approaches to approximate the true probability.

5 Performance Evaluation

Reliability plot has been used previously to measure the performance of probability estimation [16]. As a summary, we use the x-axis for the estimated probability and y-axis for the true probability. The probability estimate by a learner is good if all points are close to the $y = x$ diagonal line. Since true probability is not given for many real-world datasets, examples with similar estimated probabilities are grouped or binned together and empirical probability (that replaces true probability) is calculated by dividing the number of examples with given class label y by the number of examples in the bin. Details can be found in [16].

Due to limited number of examples, the exact shape of reliability plot depends on the chosen bin size (defined in [16]). Each curve carries no information on how many examples are in each bin. Reliability plot has no direct correlation to expected loss. **MSE** has been used previously to solve this problem [16]. Citations to earlier use of MSE can also be found in [16]. In [16], Squared error is defined as $SE = \sum_y (T(y|\mathbf{x}) - P(y|\mathbf{x},\theta))^2$, where $T(y|\mathbf{x})$ is 1 if $\mathbf{x}$ has class y and 0 otherwise. This is equivalent to assume $P(t|\mathbf{x}) = 1$ and t is the true label of $\mathbf{x}$. Under this definition, a model with 0 error rate could still have a high MSE. For example, we have a two-class problem $\{-,+\}$ and two examples, $\mathbf{x}_1$ is + and $\mathbf{x}_2$ is -. Assume that model θ_1 predicts that $p(+|\mathbf{x}_1,\theta_1) = 1.0$ and $p(-|\mathbf{x}_2,\theta_1) = 1.0$, and model θ_2 predict that $p(+|\mathbf{x}_1,\theta_2) = 0.6$ and $p(+|\mathbf{x}_2,\theta_2) = 0.6$. Obviously, if we predict the label with highest estimated probability, both models will have 0% error rate. However, under the original definition of MSE, θ_1 will have MSE at 0, and θ_2 will have MSE of 0.32. The problem comes from the strong assumption of $P(t|\mathbf{x}) = 1$. This assumption is appropriate for perfectly deterministic and noise-free problems, but is inappropriate for stochastic problems where $P(t|\mathbf{x}) \neq 1$.

In our experiments, whenever $P(y|\mathbf{x})$ is given, we set $T(y|\mathbf{x})$ to the true probability and MSE measures the exact accuracy in probability estimation. For many real-world applications, true probability is usually not known. Instead, we use MSE to measure how far the predicted probability estimate is from making the optimal decisions, as discussed in Section 3. To be specific, we use MSE to measure the difference between the estimated probability and the decision threshold to predict optimal decision $y_* = t$ (as defined in Definition 1). Assume that the decision threshold for the true class label is $v(t)$. When predicted correctly, the exact value of $P(t|\mathbf{x},\theta)$ is not important. However, when classified incorrectly or $P(t|\mathbf{x},\theta) < v(t)$, their difference $|P(t|\mathbf{x},\theta) - v(t)|$ quantifies how far the predicted probability is from making the correct decision. Intuitively, a model that is 0.1 off from $v(t)$ is better than one that is 0.9 off. Based on the original definition of MSE in [16], we propose an **improvement** that takes all these factors into account.

$$\text{Improved Squared Error} = \left(1 - [\![\frac{P(t|\mathbf{x},\theta)}{v(t)}]\!]\right)^2 \quad (3)$$

The predicate $[\![]\!]$ is defined as $[\![a]\!] = \min(1.0, a)$. Under the improved definition, a model that makes the correct prediction for $\mathbf{x}$ will always have MSE value of 0 regardless of the probability estimate. For a dataset of many examples, unless they both make no mistakes for any examples,

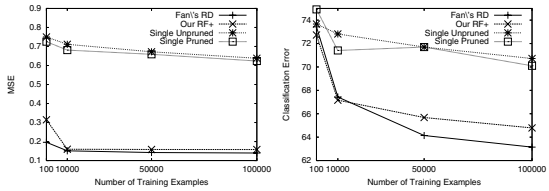


Figure 1. Synthetic Dataset: MSE and Error Rate

Table 1. Synthetic Data Bias-Variance Decomp

	Bias	Variance
Unpruned Tree	.007212	.541855
Pruned Tree	.007344	.536333
RD	.005833	.006041
RF+	.003941	.029745

two models with the same error rate may not necessarily have the same MSE. For example, suppose that two models' probability estimate for every example except for one $\mathbf{x}$ is the same, and they both make mistakes on this $\mathbf{x}$. Obviously, they have the same error rate. Assume that the true label of $\mathbf{x}$ is +. One model predict $p(+|\mathbf{x}, \theta_1) = 0.4$ and the other model predict $p(+|\mathbf{x}, \theta_2) = 0.1$. The MSE for the first model is lower than the MSE for the second one. This is because the probability estimate is closer to making the correct classification, and it exactly quantifies how one model is closer to making the correct prediction than another one.

Log-loss or **cross-entropy** has been proposed previously to measure probability estimate [16], defined as $-\sum_y T(y|\mathbf{x}) \log \frac{P(y|\mathbf{x},\theta)}{T(y|\mathbf{x})}$. We don't use cross-entroy since it is well known that cross-entropy is undefined when either $P(y|\mathbf{x}, \theta) = 0$ or $T(y|\mathbf{x}) = 0$. Importantly, the ratio $\frac{P(y|\mathbf{x},\theta)}{T(y|\mathbf{x})}$ has no direct relationship to the expected loss even if we choose $T(y_*|\mathbf{x}) = v(y_*)$.

Bias and Variance Decomposition For a given example $\mathbf{x}$, different models will normally generate different probability estimates, shorthanded as p_i. When there is no class label noise, squared error can be decomposed into bias and variance using the standard bias-variance decomposition method [3]. Bias quantifies the systematic error of a method, i.e., its average performance when the same algorithm is applied on multiple datasets, and variance quantifies the error due to variations from multiple training set. For squared error, the major prediction p_m is the mean value of p_i's. Assume that t is the true probability, then bias is $(t - p_m)^2$, and variance is $\frac{\sum (p_i - p_m)^2}{K}$. In these definitions, K is the total number of training sets and models.

6 Experiment

We use both synthetic and real-world datasets, the number of class labels is either binary or multi-class, and the loss function is either 0-1 loss or cost-sensitive loss.

Table 2. Accuracy on binary problems

donation

	unpruned	pruned	RD	RF
$	12577.61	0	14716.7	0
	RF$^+$	BT	BT$^+$	DT
$	12881.2	0	12610.5	12757.6

ccf

	unpruned	pruned	RD	RF
$	552819	659339	824024	397025
	RF$^+$	BT	BT$^+$	DT
$	813365	333409	795244	614943

adult

	unpruned	pruned	RD	RF
0-1	13733	13996	13877	13840
	RF$^+$	BT	BT$^+$	DT
0-1	13945	13903	13908	13738

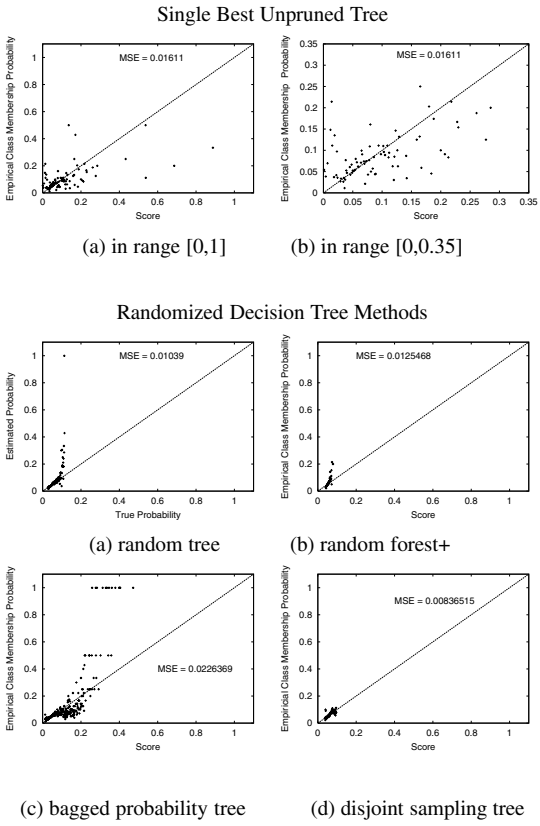
Table 3. Donation Bias-Variance Decomp

	Bias	Variance
Unpruned Tree	.19410	.153455
RDT	.17812	.031456
RF+	.18435	.056667

6.1 Synthetic Dataset

The advantage of synthetic datasets is that the true probability distribution $P(y|\mathbf{x})$ is formulated from the program script, and can be used to compute the "exact" MSE to compare results. We compute a sum from each feature multiplied by a weight, $s = a_1 \cdot x_1 + a_2 \cdot x_2 + \ldots + a_d \cdot x_d$. Each feature is within $[0, +m]$, so the maximum value of s is $S = m \cdot (a_1 + a_2 + \ldots + a_d)$. The weight adjusts the importance of a feature to calculate the class label. We compute a ratio from each feature vector and S by $\tau = \frac{s}{S}$. The class label is generated from the following probability distribution, $P(0|\mathbf{x}) = \tau \cdot (1 - \tau), P(1|\mathbf{x}) = \tau \cdot \tau, P(2|\mathbf{x}) = (1 - \tau) \cdot \tau$, and $P(3|\mathbf{x}) = (1 - \tau) \cdot (1 - \tau)$. For this dataset, it is impossible to have 0% classification error even if we know the true probability distribution. Assuming that for a particular example, τ is 0.3, and the probability distribution is $P(0|\mathbf{x}) = 0.21, P(1|\mathbf{x}) = 0.09, P(2|\mathbf{x}) = 0.21$, and $P(3|\mathbf{x}) = 0.49$. Obviously even if we know the true probability distribution, the best guess is to predict class label "3". However we are still mistaken 51% of the time when $\mathbf{x}$ is sampled repeatedly.

In our experiments, the range of each feature value is between 0 and 5, each feature value is randomly chosen between 0 and 5, and there is no dependency among the different feature values. Three dimensions $d = 5, 15$, and 20 are explored. For each dimension, a fixed test dataset with 10000 examples is used, but four significantly different training data set sizes are tested, 100, 10000, 50000,



(a) in range [0,1] (b) in range [0,0.35]

(a) random tree (b) random forest+

(c) bagged probability tree (d) disjoint sampling tree

Figure 2. donation: reliability plots

and 100000. The weight coefficient is set as $a_i = i$. The exact MSE and classification errors are in Figure 1, where the x-axis is the number of training examples and the y-axis is either MSE or classification error. The probability estimation by the two randomized decision tree approaches is not only consistently (with no exceptions among 24 tests) lower than single decision trees, but the MSE's are also significantly lower. While the MSE's of the two randomized decision trees are approximately between 0.1 and 0.3, the MSE's of the two single decision tree methods are at least 0.55. The classification errors of the two randomized decision tree methods are also consistently and significantly lower than single decision trees. The theoretically lowest classification error even if we know the true probability distribution is approximately 44%. The lowest classification error obtained by single decision trees is around 70% while the lowest error obtained by randomized methods is 65%.

Bias and Variance Decomposition In the bias and variance decomposition experiment, the fixed test set has 2000 examples, and there are 100 training sets with 2000 examples each. The results are summarized in Table 1. The error of the traditional single decision trees comes from high values in both bias and variance. The improvements by RD and RF+ are due to reduction in both bias and variance, but the reduction in variance is more significant than in bias.

6.2 Binary Problems

We have chosen three datasets donation (KDDCUP'99), credit card fraud and adult datasets that were used in previous studies [11, 10]. Detailed description about these three datasets and their loss functions (both 0-1 loss and cost-sensitive loss) can be found in [11]. The results on single decision tree and RD on these three datasets were reported earlier in [10]. However the other results are entirely new.

Accuracy The accuracy on the three chosen datasets is more straightforward than "loss", and is summarized in Table 2. Please note that donation and credit card datasets are cost-sensitive problems. The accuracy on these two datasets is total profits marked as ($). For the adult dataset, we use 0-1 loss function and the accuracy (marked as 0-1) is the number of correctly predicted examples in the test dataset. Clearly, all randomized approaches have either significantly higher or equivalent accuracy on all three data sets. The advantage of randomized approaches are particularly obvious for (cost-sensitive) donation and credit card fraud dataset. Random approaches that choose features randomly (random forest with probabilistic output and random decision tree) have significantly higher accuracy than other random approaches that choose data subset randomly. For both the donation and credit card fraud datasets, the two methods with highest profit returns are random forest with probability output and random decision tree. Please notice that for the donation dataset, any methods that directly output class label instead of probability have a zero profit since they predict everyone as non-donors.

Reliability Plots The reliability plots for single best tree and randomized decision tree methods for the donation datasets are shown in Figures 2. There are 1000 bins of equal size. The top two plots are for single best unpruned decision tree. (A pruned tree has only one node predicting everyone is a non-donor.) Since most of the points are within the range between 0 and 0.35, we "enlarged" that area on a separate plot on the immediate right. The single best unpruned tree's score scatters around the perfect matching line. However, random forest, random decision tree and disjoint sampling tree match the true probability very well. To summarize these results, we use MSE to measure how closely the score matches the empirical probability. In other words, v used in Eq 3 is the empirical probability measured from the testing data. The MSE for single best unpruned tree is 0.01611 while the MSE for randomized decision tree methods (except for bagged decision tree) is at most 0.0124. The bagged decision tree has higher MSE than the single unpruned tree. We look at the predictions and find that this is due to the fact that the percentage of positives is around 5% and every tree trained from each bootstrap is highly correlated on the 95% negatives.

Bias and Variance Decomposition The bias and variance decomposition on the donation dataset is summarized in Table 3. The original test dataset is chosen, and 100 training sets with 10000 examples each are sampled from the original training set. The true probability is the empirical probability. Similar to the synthetic datasets, the reductions by RD and RF+ come from both bias and variance, but mainly in variance.

6.3 Multi-class Problems

Data Sets The artificial character dataset from UCI has been artificially generated by using a first order theory which describes the structure of ten capitol letters of the English alphabet and a random choice theorem prover which accounts for heterogeneity in the instances. The capitol letters represented are the following: A, C, D, E, F, G, H, L, P, R. There are 1000 instances (100 per letter) in the training set, and 5000 instances (500 per class) in the test set.

Reliability Plots The reliability plots are per class, as shown in Figure 3. Since the dataset is relatively small considering there are 10 class labels, we have chosen to divide the range into 10 bins. A single decision tree may not predict a posterior probability in every bin since it only has a limited number of leaf nodes, and each node can output just one probability number. When we draw the curve for each method, we only use the line to connect immediately adjacent bins. If a model predicts in in bin 1, 2, 5 and 6, we connect bin 1 and 2, break between 3 and 4, and connect 5 and 6 again. In general, RD appears to be very systematic in every single class of all 10 classes. For all 10 plots, RD's reliability plot is always continuous, monotonically increases and it mostly stops at scores between 0.3 and 0.4. Except for "Letter L" of the artificial character dataset, RD can detect all positives with no false positives at the last bin. In contrast, the reliability plot of the single unpruned tree is obviously very un-systematic and varies significantly from class to class for both datasets. Unlike RD, the plots for most classes are not continuous. In many cases (letters A, C, E, F, G), the reliability plots are not monotonically increasing, i.e., higher predicted probability actually have a lower empirical probability measured from the data. RF+'s reliability plots have characteristics from both RD and single unpruned tree. They are more systematic than the single decision tree, and there is clearly a monotonically increasing trend for all plots except for letters E and F. Between RF+ and unpruned best tree, RF+ reliability plot matches the unpruned tree in a number of cases (letter D, F, and L) Each tree of RF+ is trained from the same training set and uses the same "information gain" selection criteria as the single unpruned tree. There is some extent of correlation between RF+ and single unpruned tree. Comparing with RD, RF+ predicted probability ranges from 0 to 1, while RD's predicted probability ranges from 0 to 0.4. This is due to the bias of the chosen hypothesis space Θ. The hypothesis space of RD are the trees whose feature at each node can be any remaining feature and the depth is limited to the number of features. Under this inductive bias, most leaf nodes are not pure, i.e., they contain a mixture of examples belonging to many different classes. However, RF+ chooses features with "information gain" and no depth limitation on pruning. Consequently, the leaf nodes of RF+ trees are much purer. To verify this, we include the prediction of an example from the dataset.

	A	C	D	E	F	G	H	L	P	R
RD	.0285	.0321	.0676	.0544	.0660	.0279	.0231	.0514	.302	.347
RF+	0.0	0.0	0.0	0.0	.00399	0.0	0.0	0.0	.282	.714
up	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	.128	.872

The true class label for the example is letter R and all three models have the correct prediction. Interestingly however, all three models think that there is also some probability that this example could be letter P. Indeed, P and R do look similar. Clearly, RD's prediction has none-zero probability for each of the 10 classes and the one with the highest probability is the true label. The top three classes that RD predicts to be closest to the true label are R, P and F. On the other hand, RF+ has none-zero probability for only the three letters R, P and F, which are exactly the top three possibilities predicted by RD. However, the unpruned single tree has a clean cut prediction that the choice is either P with .872 probability or R with .128 probability.

Error Rate and MSE Error rate and MSE results are in Table 4. We measured prediction error rate with three different criteria. Top 1 means that if the true label of $\mathbf{x}$ is the one with the highest estimated probability, the prediction is considered correct. Top 2 means if the true class label of $\mathbf{x}$ is among the two class labels with the highest predicted probability, we consider $\mathbf{x}$ is predicted correctly. Top 3 is similar to top 2. For example, if the true class label of $\mathbf{x}$ is R and the three class labels with the highest predicted probabilities are F,R and P (and their predicted probabilities are .0606, .302 and .349 respectively), the prediction is considered correct under both top 2 and top 3 criteria, but regarded as a mistake under top 1 criterion. For Table 4, RD has an error rate of 14.58% under the top 1 criterion, while RF+'s error rate is 21.82%. The error rate of the two single decision trees are around 27.7%, which are much higher than both RD and RF+. When the criterion becomes less restrictive, i.e., top 2 and top 3, the error rates of RD and RF+ decrease significantly and drop down to 0. However, for the two single decision trees, even after the criteria are less restricted, the error rate still remains above 7%. In particular, the error rate at top 3 is exactly the same as top 2. The single decision trees' leaf nodes have examples of at most 2 classes and can only predict none-zero probability for two classes.

We have computed four different measurements of MSE according to Eq 3 and the results are in the bottom table of Table 4. The first row with leading column of "1.0" uses 1.0 as the decision threshold. This is the same measure used in

Table 4. Error rate and MSE for the Artificial Char

Error Rate

	unpruned	pruned	RD	RF+
top 1	27.78%	27.74%	14.58%	21.82%
top 2	8.32%	7.38%	0.26%	0.02%
top 3	8.32%	7.38%	0%	0%

MSE

	unpruned	pruned	RD	RF+
1.0	0.439	0.435	0.734	0.374
top 1	0.360	0.356	0.0387	0.181
top 2	0.288	0.269	0.00442	0.00106
top 3	0.288	0.269	0	0

Table 5. Aritifical Char Bias-Variance Decomp

	Bias	Variance
Unpruned Tree	.31240	.156134
Pruned Tree	.31425	.149244
RD	.04133	.002347
RF+	.17714	.092458

[16]. The other rows in the MSE tables, i.e., top 1 to top 3, are the MSE measures with different decision thresholds. The decision threshold used by top 1 is the highest predicted probability. The decision threshold for top 2 and top 3 are the 2nd and 3rd highest predicted probability respectively. It is important to understand that the actual decision threshold chosen for each example is different, and it is different for different methods. Each example is predicted with different probabilities and the prediction for the same example is different for different methods. The MSE measure is only meaningful to understand each method's probability estimation towards achieving 0 error rate under the given criterion. A higher MSE means that on average the predicted probabilities for true labels are more off from the decision threshold for correct classification.

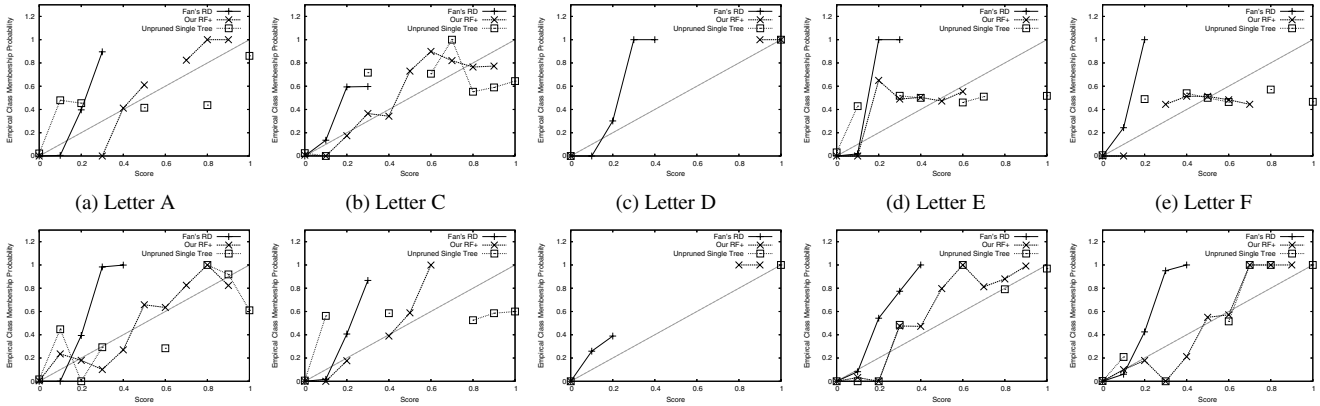
Bias and Variance Decomposition We used the original test set of 5000 examples and randomly sampled 100 training sets with 250 examples each. Top 1 decision threshold is used as the true probability. The bias and variance decomposition results are summarized in Table 5. Similar to the bias and variance decomposition result of the synthetic dataset in Table 1, the reduction in MSE comes from both bias and variance. The reduction in variance is significant for both RD and RF+. The reduction in bias by RD is significantly more than by RF+.

7 Related Work

Two important works in randomized decision trees that are "seemingly" not experimented within our study are Amit and Geman's "randomized trees" [1] as well as Dietterich's "randomly choose one of the top k attributes" [9]. In their randomized trees, Amit and Geman randomly generates feature subset from the complete feature set; from each feature subset, they run a conventional decision tree learner to compute the single best tree. Dietterich's "randomly choose one of the top k attributes" method randomly selects one of the feature among the top k with highest information gain. The spirit of both approaches is incorporated in Breiman's random forest [6], which is tested in our experiments. In an earlier work [2], Bauer and Kohavi evaluated a few voting classification algorithms including bagging, boosting and arcing. The major differences in our study from this early work are as follows. Our base level decision trees output probabilities while they evaluated methods that output class labels. The combination method in this paper is to "average multiple probability outputs" rather than "choose a class label with the highest (either simple or weighted) vote." Probabilistic output is more flexible and easier to use under a variety of loss functions. Regarding algorithmic differences, our paper evaluated random decision tree, random forest and its probabilistic variation, probabilistic variation of bagging, and disjoint sampling tree, which were not covered by their work. Additionally, our work concentrates on the ability to match the posterior probability, while their earlier work concentrates on minimizing 0-1 loss. In [15], the decision tree algorithm has been extended to provide reliable probability-based rankings of multiple classes. In [10], RD itself has been evaluated against the three binary datasets. In our paper, we consider a whole family of randomized decision tree methods not only RD, but also random forest plus, bagged probability tree and disjoint sampling tree. In addition, their various mechanisms are explained using Bayesian optimal classifier to justify our claim that their actual mechanism is to estimate the true posterior probability of the target function using the decision tree space. A problem with previously proposed MSE has been corrected. A significantly more extensive empirical study including deterministic and stochastic datasets, and large multi-class problems was conducted. Our paper makes much wider and stronger claims than [10]. In [8], Chipman et al. proposed several approaches, particularly Monto-Carlo Markov Chain approach, to set the prior and posterior probabilities to more sophisticated randomized approaches to construct random decision trees, i.e., randomly choosing one of the following four procedures, Grow, Prune, Change, and Swap. The major distinction is that the approaches explored in this paper employ rather simple uniform prior and posterior probability assigment as well as simple approaches to grow random trees.

8 Conclusion

We discussed both traditional decision tree and four randomized decision tree algorithms as estimators to true posterior probability. These four algorithms cover many randomization techniques and ideas independently proposed by several researchers, ranging from randomly-structured to



(a) Letter A (b) Letter C (c) Letter D (d) Letter E (e) Letter F

(f) Letter G (g) Letter H (h) Letter L (j) Letter P (j) Letter R

Figure 3. artificial character: reliability plots of single unpruned tree, RD and RF+

carefully-structured trees, and from original training set to randomized training set. We established the relationship between accurate posterior probability estimation and reduction in expected losses. We proposed an improved definition of MSE to measure the effectiveness of probability estimation to minimize a given loss function. Experimental studies have found that the estimated probabilities by the four randomized decision tree methods appear to be very systematic and monotonically increasing, as shown in reliability plots. However, the reliability plots of single decision trees appear to be sporadic and have no clear increasing trends. The estimated probability by randomized decision methods are well correlated with true probability. The improved MSE definition effectively measures a model's ability to minimize expected loss. The MSE measurements for the same datasets are significantly lower for the randomized decision tree approaches than for single decision trees. The bias and variance decomposition shows that the reduction of MSE by randomized decision tree methods is in both bias and variance, although the reduction in variance is more significant. We also find that the probability estimates of different randomized decision trees have different behaviors, such as the difference in predicted value range between RD and RF+.

Future Work The results in this paper have shown that minimizing error in probability estimation can effectively reduce both 0-1 and cost-sensitive losses. Though estimating posterior probabilities directly is traditionally regarded as a more difficult problem than directly predicting class labels, our work shows that predicting reliable probabilities with the family of randomized decision trees appears straightforward and accurate. In the future, we are interested in looking for a new method that explores a different hypothesis space than any of the existing randomized decision approaches, and ideally hope to estimate probability better. From the analysis of RD and RF+ in bias-variance decomposition and range of predicted values, it appears that some kind of their combination may be a good avenue to explore.

References

[1] Yali Amit and Donald Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9(7):1545–1588, 1997.

[2] Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning*, 36(1-2):105–139, 1999.

[3] E Bienenstock, S Geman, and R Dorsat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[4] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[5] Leo Breiman. Randomizing outputs to increase predictiong accuracy. *Machine Learning*, 40(3):229–242, 2000.

[6] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[7] Wray Lindsay Buntine. *A Theory of Learning Classification Rules*. PhD thesis, School of Computing Science, University of Technology, Syndney, 1992.

[8] H Chipman, E George, and R McCulloch. Bayesian CART model search. *J of American Statistics*, 93(98):935–960, 1998.

[9] Thomas Dietterich. An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization. *Machine Learning*, 40(2):139–157, 1998.

[10] Wei Fan. On the optimality of probability estimation by random decision trees. In *Proceedings of Nineteeth National Conference on Artificial Intelligence (AAAI 2004)*, pages 336–341, San Jose, CA, July 2004.

[11] Wei Fan, Haixun Wang, Philip S Yu, and Sheng Ma. Is random model better? on its accuracy and efficiency. In *Proceedings of Third IEEE International Conference on Data Mining (ICDM-2003)*, Melbourne, FL, Nov 2003.

[12] Jennifer Hoeting, David Madigan, Adrian Raftery, and Chris Volinsky. Bayesian model averaging: A tutorial. *Statistical Science*, 14:4:382–401, 1999.

[13] Fei Tony Liu, Kai Ming Ting, and Wei Fan. Maximizing tree diversity by building complete random decision trees. In *Proceedings of Nineth Pacific Asian Knowledge Discovery and Data Mining Conference (PAKDD'05)*, May 2005.

[14] Tom M. Mitchell. *Machine Learning*. McGraw Hill, 1997.

[15] Foster Provost and Pedro Domingso. Tree induction for probability-based ranking. *Machine Learning*, pages 199–215, September, 2003.

[16] Bianca Zadronzy and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of Eighth International Conference on Knowledge Discovery and Data Mining (KDD-2002*, Edmonton, Alberta, Canada, August 2002.