A Comparative Analysis of Methods for Probability Estimation Tree

NA CHU, LIZHUANG MA
Department of Computer Science & Engineering
Shanghai Jiao Tong University
No. 800, Dongchuan Road, Shanghai
P.R.CHINA
cina19@163.com; ma-lz@cs.sjtu.edu.cn http://www.sjtu.edu.cn

PING LIU, YIYANG HU, MIN ZHOU
Institute of Liver Diseases
Shanghai University of Traditional Chinese Medicine
No. 1200, Cailun Road, Shanghai
P.R.CHINA
liuliver@online.sh.cn; yyhu@citiz.net; zzz208216@hotmail.com
http://www.shutcm.com

Abstract: - In this paper, we address the problem of probability estimation of decision trees. This problem has received considerable attention in the areas of machine learning and data mining, and techniques to use tree models as probability estimators have been suggested. We make a comparative study of six well-known class probability estimation methods, measured by classification accuracy, AUC and Conditional Log Likelihood (CLL). Comments on the properties of each method are empirically supported. Our experiments on UCI data sets and our liver disease data sets show that the PETs algorithms outperform traditional decision trees and naïve Bayes significantly in classification accuracy, AUC and CLL respectively. Finally, a unifying pseudocode of algorithm is summarized in this paper.

Key-Words: - Probability estimation tree, Decision trees, Classification, Joint distribution, AUC, Conditional log likelihood

1 Introduction

Decision trees, as the classification algorithm, have been studied in detail both in the areas of machine learning and data mining. Several factors contribute to its popularity. They are praised for comprehensibility of their knowledge representation and inference procedures, in contrast to neural networks. They are also non-parametric, which have facilitated their wide use in the comparison of different learning algorithm [1]. They can treat comparatively well with large scale applications [2].

As they have been used in most research and applications, decision trees are a way to represent rules underlying data with hierarchical, sequential structures that recursively divide-and-conquer partition the data. Various algorithms have been developed for learning decision trees. Among them, the C4.5 algorithm of Quinlan is often used, which evolve from an algorithm, called ID3 [3]. The C4.5 algorithm uses a greedy search, and searches through the attributes of the training instances and extracts the attribute that best separates the given samples, and results in crisp decisions at leaf nodes, and the aim is for high classification accuracy.

However, crisp decisions that decision trees usually output may not be adequate or desirable in some practical applications, such as medical diagnosis, the value of predicting class probability of diagnosis may be important for patients. For example, suppose we are estimate the probability of heart disease given blood pressure. Surely we should predict a higher probability of heart disease for the patient with blood pressure of 250 than for the patient with 160.

From this view, it is clear that ranking samples by the class probability are more essential than class predictions. Many methods have been proposed for building probability estimation trees (PETs), the best known assigning a probability distribution for all classes at the leaf nodes.

There are three main problems in building class probability trees. Firstly, the greedy top-down construction is the most commonly used method for tree growing today. Existing probability estimation tree algorithms estimate probability separately in each leaf, similar to a traditional decision tree, so building a probability estimation trees is a greedy and recursive process.

The second of which concerns how to represent accurate probabilities but also to be easily learnable from limited data in practice. Traditional decision trees, such as C4.5, have been observed to produce poor probability estimation, even though to produce the better classification accuracies [4]. Provost and Domingos [5] point out that the reason behind the poor estimates of decision trees is not the decision tree representation, but the inductive algorithm.

The third problem, which Han Liang et al. [6] deem more important, concerns the evaluation metrics. According to [25], the ACC can not provide solid evidence by itself for probability estimation models. In this paper, we will apply three different ways to evaluate the probability estimation of a learning model: ACC, AUC and CLL.

- ACC is the classification accuracy, and calculated as the percentage of the correctly classified testing samples over all the test samples.
- AUC is the area under the Receiver Operating Characteristic curve, and is a relative evaluation standard, and has been recently proposed as an alternative single-number measure for evaluating the predictive ability of learning algorithms. It can be easily calculated for a binary-class problem as follows:

$$AUC = \frac{S_0 - n_0(n_0+1)}{n_0 n_1} \qquad (1)$$

Where n_0 and n_1 are the numbers of the positive and negative test samples respectively, and S_0 is the sum of the ranks of the positive test samples. For a multi-class problem, AUC is calculated by M-measure in [7]:

$$M = \frac{2}{n_c(n_c-1)} \sum_{i<j} AUC(i,j) \qquad (2)$$

where n_c is the number of class.

- CLL is the Conditional Log Likelihood, and is a more straightforward measurement to evaluate learning models with respect to probability estimation, and describes the reliability of probability estimation. The formal CLL definition is given as follows[6, 8]:

$$CLL = \sum_{t=1}^{n} \log \widehat{P}(C \mid s_t) \qquad (3)$$

where $\widehat{P}(C \mid s_t)$ is the conditional probability of C given a test sample s_t.

In general, the PET methods count the proportions from each class which are present at the leaf nodes, based on the training data, and generate a local maximum likelihood estimate or perhaps a smoothed variant of posterior class probabilities [9]. Their beneficial effects have seen increasing use in many applications [10, 11, 12], and their attractive properties have attracted the attention of many researchers, who have proposed a number of methods.

The main objective of this paper is to make a comparative study of some of the well known PET methods with the aim of understanding their theoretical foundations, their strengths and the weakness. Naïve Bayes classifiers are generally easy to understand when the log probabilities were presented as evidence that adds up in favor of different class. Therefore, in this paper, we compare C4.5 tree algorithm, Naïve Bayes algorithm with other four popular PETs with respect to class probability estimation, measured by ACC, AUC and CLL,and the results are showed in Section 3. Those four algorithms include C4.4, NBTree, CITree and CLLTree. In Section 2, we are devoted to a critical review of four PETs algorithms which have achieved wide-spread popularity, while Section 3 provides empirical support for some comments by examined on ten real-world datasets. Finally, in Section 4, we conclude with discussion and some directions for future research.

2 A Critical Review of PET

We denote a vector of attributes by an upper-case letter A, $A = (A_1, A_2, \cdots, A_n)$, and an assignment of value to each attribute in A by a corresponding lower-case letter a. We use C to denote the class variable and c to denote its value. Therefore, a training sample $s = (a, c)$, where $a = (a_1, a_2, \cdots, a_n)$, and a_i is the value of attribute A_i.

In PETs, the class probability $\widehat{p}(c \mid s)$ denotes that a sample s is classified into the class c with the class probability $\widehat{p}(c \mid s)$, and which is estimated by the fraction of the samples of the class c in the leaf into which s falls.

In the following subsections, we will review the existing work on augmenting decision trees to estimate precise class probabilities. And we will briefly outline these methods according to their construction methodologies and use them, and comment the strengths and weaknesses of each method.

2.1 C4.5 and C4.4

2.1.1 Description

The C4.5 method (with pruning), proposed by Quilan [3], has been found to provide poor probability estimates. There are two obstacles in building an accurate probability estimate of the C4.5 tree. One of which is the class probabilities of all the test samples in the same leaf are equal, which prevents accurate probability estimation. The other obstacle is that C4.5 is biased towards building small trees with fewer leaves because of post-pruning [5]. Provost and Domingos [5, 13] therefore develop a C4.4 algorithm to improve probability estimation of C4.5 decision trees.

The C4.4 can be built by modified the C4.5 in two ways:

- Turn off the pruning and collapsing. The pruning and collapsing can remove those branches which may be useful for probability estimations but not improving resultant accuracy on a test samples. It means that pruning damages the probability estimation of traditional decision trees. Thus, to build accurate PETs it should not to prune and collapse at all.
- Smooth probability estimates by the Laplace correction. To avoid producing probabilities of extreme values (e.g., 100% or 0%), they use the Laplace correction to smooth the estimation and make it less extreme. Assume there are samples that have the class label at a leaf, total samples, and class values in a sample set. Thus the Laplace estimation calculates the estimated probability as follows:

$$\hat{p}(k \mid s) = \frac{n_k + 1}{N + C} \qquad (4)$$

2.1.2 Comments

The Laplace correction and turning off pruning and collapsing result in generating larger trees to give more precise probability estimation. As we will see later, the better performance of C4.4 can be showed by our experiments.

However, the large tree may overfit the training samples so that the probabilities estimated may not be accurate. In addition, when the depth of the tree is large, there is very small number of training samples with each leaf node. Thus, the probability estimates are less reliable [14]. This problem is also particularly noticeable. Moreover, there are still many duplicate class probability values, which can substantially decrease the quality of ranking test samples based on their class probabilities.

2.2 Naïve Bayes Tree (NBTree)

2.2.1 Description

The NBTree, proposed by Ron Kohavi [15], is a hybrid approach that utilizes the advantages of both decision trees and Naïve-Bayes.

In building NBTree, the algorithm similar regular decision trees (e.g. C4.5) to recursively partition the sample space according to the best attribute, except that after a tree is grown, a Naïve Bayes classifier is constructed at the leaves using the samples associated with those leaves. The split score function of selecting the best split attribute is classification accuracy.

According to the Bayes theorem which requires making strong independence assumptions the NBTree uses the method of maximum posterior class probability to denote probability estimation, $p\left(c_j \mid s_t\right)$ as bellows:

$$\hat{p}\left(C \mid A_p(L)\right) = \frac{\hat{P}(A_p(L) \mid C)\hat{P}(C)}{\hat{P}(A_p(L))} \qquad (5)$$

Since $\hat{P}(A_p(L))$ is a constant independent of C, then the test sample s_t is classified into class c can be get from the following equation.

$$C_{\text{lnb}} = \arg\max \hat{P}(A_p(L) \mid C)\hat{P}(C) \qquad (6)$$

where $A_p(L)$ are the attributes that occur in the path from the root to a leaf L.

2.2.2 Comments

In building NBTree, Kohavi validated a split by estimating the reduction in error, which is gained by the split and comparing it to a predefined threshold of 5%. At the same time, he limits at least 30 samples at the current node. The reason is that splitting a node with only a few training samples will seriously affect the final accuracy and will lead, on the other hand, to a complex and less comprehensible decision tree.

According to the simple restrictive conditions that after growing a NBTree, then the number of nodes induced by the NBTree is in most cases smaller than that of C4.5. However, when testing the significance of a split in a node, the inner v-fold cross-validations accuracy procedure is used. Due to the cross-validation estimations, the NBTree becomes computationally expensive for methods that are more time-consuming than naive Bayes, although the NBTree applies a Naive Bayes classifier to decision tree leaf nodes, and it outperforms C4.5 and naïve Bayes [16].

2.3 Conditional Independence Tree (CITree)

2.3.1 Description

Harry Zhang and Jiang Su [17, 18] extended decision trees to represent a joint distribution and conditional independence, called conditional independence trees (CITree). This approach attempts to iteratively explore and represent conditional attribute independencies at each step in constructing a decision tree, and thus in learning a CITree, they want to select the attribute that make local conditional independence among other attributes true as much as possible at each step, and means that the attribute given all other attributes has the maximum conditional independence. For this reason, unlike the NBTree, after growing a CITree, given the attributes that occur on the path from the root to a leaf, all the other leaf attributes are independent. So the probability estimates in a leaf can be given by

$$\widehat{p}\left(A,C\right)=\widehat{p}\left(C\mid A_p(L)\right)\widehat{p}\left(A_l(L)\mid A_p(L),C\right) \quad (7)$$

Where $A_p(L)$ are the attributes that occur in the path from the root to a leaf L, and $A_l(L)$ are all the attributes not in $A_p(L)$.

- $\widehat{p}\left(C\mid A_p(L)\right)$ represents the conditional independence distribution at a leaf L.
- $\widehat{p}\left(A_l(L)\mid A_p(L),C\right)$ is local conditional distribution which is the conditional probability distribution over the leaf attributes at each leaf.

From the conditional independence assumption of naïve Bayes, the following equation stands:

$$\widehat{p}\left(A,C\right)=\widehat{p}\left(C\mid A_p(L)\right)\prod_{i=1}^{m}\widehat{p}\left(A_{li}(L)\mid A_p(L),C\right) \quad (8)$$

Where m is the number of attributes at a leaf node.

2.3.2 Comments

Buiding a CITree is also a greedy and recursive process. The split score function of selecting the best split attribute is also classification accuracy. The difference of C4.4, NBTree and CITree is that the formers represent the conditional probability distribution of the path attributes, while the CITree represents a joint distribution over all the attributes. According to their experiments in [17, 18, 19], CITrees demonstrate good performance in both classification and ranking. However, learning a CITree tends to have relatively higher computational complexity compared with learning a traditional decision tree.

In addition, Harry Zhang and Jiang Su also discuss that the average size of CITrees is much smaller than that of C4.4 and NBTree over most of data sets [17, 20].

2.4 Conditional Log Likelihood Tree (CLLTree)

2.4.1 Description

Similar to the NBTree, in each step of expanding the decision tree, the Conditional Log Likelihood (CLL) [21] is used as the splitting criterion or score function to select the best attribute to split. The splitting process ends when two conditions are met. One of conditions is at least 30 training samples at the current leaf. The other is the relative reduction in CLL is greater 5% when comparing two alternatives in terms of CLL value, which are calculated by a cross-validation procedure. Finally, for the samples at leaves, naive Bayesian models are generated, which optimizes the estimation of class posterior probability.

The conditional probability of a sample in the CLLTree method can be represented as follows [21, 22]:

$$\log\left(\widehat{p}(C\mid A)\right)=\log\left(\widehat{p}(C\mid A_{l,}A_p)\right) \quad (9)$$

$$=\log\left(\widehat{p}(C\mid A_p)\right)+\log\left(\widehat{p}(A_l\mid C,A_p)\right)-\log\left(\widehat{p}(A_l\mid A_p)\right)$$

Then the CLL of a CLLTree is

$$CLL\left(\Gamma\mid S\right)=\sum_{i=1}^{n}\log\widehat{P}_{\Gamma}\left(C\mid A\right) \quad (10)$$

where Γ is a learning model (e.g., CLLTree), and S is a (sub) sample set with n samples.

2.4.2 Comments

The positive property of this method is that it intends to choose the attributes that maximum the posterior class probabilities among the training samples at the leaf as much as possible. Thus, even though there may exist a high impurity at its leaves, it could still be a good CLLTree.

Another positive property of this method is that the CLLTree algorithm outperforms or competitive with the state-of-the-art other above PETs learning algorithms in both classification and ranking according to the author's experiments. On the other hand, the CLLTree algorithm is also a greedy and recursive algorithm, and the time-complexity is equivalent to the NBTree.

3 Empirical Comparisons

3.1 The Design of the Experiment

In this section, we present the results of an empirical comparison of the methods (C4.4, NBTree, CITree and CLLTree) presented above with C4.5 with Laplace correction (C4.5-L) and Naïve Bayes (NB).

The main characteristics of the data sets considered in our experiments are reported in Table 1. Our experiments are conducted on the basis of 8 UCI data sets which are all publicly available at the UCI Machine Learning Repository [23], and 2 Traditional Chinese Medicine (TCM) data sets which are obtained from Shanghai University of Traditional Chinese Medicine and Institute of Liver Diseases, Shanghai, China. The two hepatitis sample sets is actually the union of five data sets on hepatitis, with the same number of attributes but collected in five distinct hospital(Shuguang, Longhua, Yueyang and Putuo Central hospitals, as well as Infectious diseases hospital, Shanghai, China). The hepatitisS sample set involves posthepatitc cirrhosis patient's 67 TCM symptoms, one TCM syndrome which were pre-classified into four classes, blood stasis-heat accumulation (88 cases), internal accumulation of damp-heat (101 cases), liver-kidney yin deficiency (41 cases) and liver stagnation and spleen asthenia (38 cases). The hepatitisW sample set describes cirrhosis disease. Moreover, samples have been assigned to two distinct classes: compensated (188 cases) and decompensated (252 cases) given 63 TCM symptoms.

In Table 1, the column head "real" concerns the number of attributes that are treated as real-value attributes. Those real-value attributes are discretized by the supervised MDLP [24] discretization method. In the column "Missing", we simply report the presence of missing values in at least one attribute of any observation. The missing values of real-value attributes are replaced by the mean value, and the missing values of categorical attributes are replaced by the nodes.

In our experiments, each data set is randomly divided into ten times into 90 percent of the samples for training and 10 percent for testing according to ten-fold cross validation. We run C4.5-L, Naive Bayes, C4.4, NBTree, CITree and CLLTree on the same training sets and test them on the same testing sets to obtain the ACC, AUC and CLL scores. The experiment results presented are averages of these 10 runs.

To study the performance of PETs methods, we compare the classification accuracy (ACC), AUC and CLL values of PETs methods with those of the corresponding traditional decision tress, such as the C4.5 with Laplace correction. In addition, we also compare the sizes of trees among them.

Table 1. Description of data sets used in our experiments.
Datasets above the horizontal line are from UCI and those below are from TCM.

Data Set	Classes	Size	Attribute	Real	Missing
Breast	2	699	10	0	Y
Breast(W)	2	569	31	30	N
Chess	2	3169	36	0	N
Dermatology	6	366	34	1	Y
Heart	2	270	14	5	N
Mushroom	2	8124	23	0	Y
SPLICE	3	3190	62	0	N
Vote	2	435	17	0	Y
HepatitisS	4	268	68	0	N
HepatitisW	2	440	63	0	N

3.2 Experimental Results

In this section, the results of different experiments of PETs methods on various data sets are summarized and discussed.

Table 2, Table 3 and Table 4 show the experimental results in terms of ACC, AUC and CLL respectively on ten datasets.

The first factor that we analyze in this section is the classification accuracy. The performance of a traditional decision trees is a usually measured by its classification accuracy on testing samples. The experiment (see Table 2) have been conducted and published on comparing, in terms of accuracy, C4.5 with pruning and Laplace correction, Naïve Bayes and several PETs models presented above. According to the average classification accuracy (see Table 2), the C4.4 achieves the highest ACC among all learning models. However, as we can see from the Table 5, the C4.4 sacrifices its tree size to improve the classification accuracy. The reason is that a larger tree easy to yield better fits a smaller tree. In addition, we notice that the classification accuracies of the NBTree, CITree and CITree models are all better than that of C4.5-L. Therefore, it is verify empirically that the details of the growing phase are less critical to obtaining good PETs than the choice of pruning mechanism [5].

However, from the outcomes of tests in term of AUC reported in Table 3, it is worthwhile noting, for most datasets, that the C4.4 outperforms the C4.5-L but worse than other PETs models in term of AUC. This is because that the accuracy measure does not consider the probability of the prediction. Thus ACC is not enough for the performance of evaluation of PETs models. Moreover, the C4.4 produces a large amount of repeated class probabilities at leaves, which greatly degrades its class probability-based ranking quality.

In addition, as a combination of a decision tree and naïve Bayes, NBTree and CLLTree are bettern than C4.5-L and NB in AUC and ACC, and CITree is second to NB in AUC. Since PETs models have explored the conditional independence among attributes in building trees. Thus, the class probability estimates of the PETs models are expected to be more accurate than those of naïve Bayes in AUC.

Thirdly, as two aspects of probability estimation, CLL and AUC represent the reliability and resolution of PETs learning models respectively [5]. In our experiments, Table 4 shows the experimental results in terms of CLL. In terms of the average CLL, CLLTree achieves the highest CLL among all algorithms.

As an extension of NBTree, the CLLTree outperforms NBTree in terms of CLL and AUC significantly, and also slightly better in ACC. This verifies results of previous publications [21, 22]. Similarly, the NBTree outperforms NB in terms of CLL, AUC as well as ACC. The CLLTree and NBTree define a probability density estimator at leaves. Obviously, such an estimate can improve the quality of probability estimation.

Finally, as expected, the average sizes of the grown trees induced by PETs except C4.4 are smaller than that of C4.5. It verifies that the PETs is often much more compact than a traditional decision tree. Among PETs, The sizes of CLLTrees are significantly smaller than the sizes of other trees over all the datasets. The size of CITree is second to the size of CLLTree. The size of NBTree is the larger than that of CLLTree and CITree. Therefore, we conclude that a full join distribution at a leaf is more compact than conditional probability distribution at a leaf. Here the size of a tree is the number of leaves.

Table 2. Experimental results for several PETs models: ACC & standard deviation

Data Set	C4.5-L	NB	C4.4	NBTree	CITree	CLLTree
Breast	92.70±1.06	97.28±0.75	94.27±0.06	96.70±0.07	96.15±0.08	95.08±0.08
Breast(W)	95.78±0.02	95.60±0.04	95.95±0.02	96.13±0.03	93.69±0.03	94.00±0.07
Chess	98.31±0.37	87.85±1.91	99.41±0.06	94.92±2.05	96.86±0.87	98.93±0.65
Derma.	93.71±0.06	97.54±0.05	94.26±0.04	97.26±0.04	97.86±0.03	93.33±0.09
Heart	77.77±2.42	83.07±2.70	78.88±2.86	81.85±2.50	84.81±2.71	84.07±2.23
Mushroom	100±0.00	95.15±0.78	100±0.00	100±0.00	93.76±0.50	100±0.00

Table 2. (Continued)

Data Set	C4.5-L	NB	C4.4	NBTree	CITree	CLLTree
SPLICE	94.35±0.07	95.33±0.14	92.66±0.04	95.33±0.02	95.16±0.02	94.92±0.05
Vote	96.32±0.02	90.11±0.05	94.94±0.03	94.29±0.03	90.00±0.07	92.71±0.07
HepatitisS	65.28±0.17	63.77±0.13	80.32±0.13	62.58±0.06	83.69±0.03	68.29±0.09
HepatitisW	73.63±0.15	75.45±0.44	85.68±0.09	72.50±0.10	60.91±0.12	77.36±0.10
average	88.78	88.11	91.81	89.16	89.29	89.56

Table 3. Experimental results for several PETs models: AUC & standard deviation

Data Set	C4.5-L	NB	C4.4	NBTree	CITree	CLLTree
Breast	97.40±0.7	99.3±0.82	97.8±0.16	99.1±0.11	97.36±0.13	98.64±0.12
Breast(W)	94.6±0.42	99.1±0.51	98.1±0.39	98.8±0.33	95.83±0.30	96.46±0.32
Chess	95.64±0.16	95.2±1.19	99.9±0.14	99.6±0.13	94.75±0.14	99.82±0.20
Dermat.	77.7±0.11	79.9±0.24	78.9±0.09	85.69±0.10	82.95±0.10	86.29±0.11
Heart	83.7±0.61	91.00±0.25	88.20±0.16	88.6±0.11	86.47±0.12	94.49±0.13
Mushroom	100±0.00	99.70±0.07	100±0.00	100±0.00	100±0.00	100±0.00
SPLICE	97.7±0.1	99.30±0.27	98.1±0.1	99.3±0.31	99.36±0.1	99.45±0.27
Vote	96.3±0.16	96.10±0.39	96.9±0.13	98.6±0.16	98.06±0.15	98.96±0.20
HepatitisS	76.20±0.07	85.20±0.10	81.70±0.06	89.30±0.06	89.83±0.30	88.27±0.06
HepatitisW	70.00±0.28	78.40±0.95	73.40±0.23	77.6±0.25	80.35±0.28	82.51±0.29
average	88.92	92.32	91.30	93.65	92.12	94.49

Table 4. Experimental results for several PETs models: CLL & standard deviation

Data Set	C4.5-L	NB	C4.4	NBTree	CITree	CLLTree
Breast	-12.10±4.64	-18.28±14.16	-11.17±3.39	-12.84±9.33	-14.94±8.57	-11.43±6.38
Breast(W)	-9.40±4.89	-28.57±8.03	-22.08±6.77	-12.96±0.81	-12.54±8.89	-10.35±2.89
Chess	-62.82±18.01	-93.48±7.65	-14.08±9.46	-54.66±27.15	-58.54±22.98	-44.09±16.62
Derma.	-32.65±20.69	-180.36±60.53	-28.93±16.01	-23.64±11.71	-18.98±10.53	-13.07±6.94
Heart	-13.09±3.49	-12.25±4.96	-14.38±6.25	-14.41±5.78	-11.23±8.03	-10.32±3.2
Mushroom	-2.10±0.19	-105.77±23.25	-2.10±0.19	-0.14±0.14	-2.52±0.23	0.00±0.01
SPLICE	-85.73±49.98	-46.53±12.85	-112.46±67.16	-57.33±24.50	-60.98±33.70	-54.66±27.15

Table 4. (Continued)

Data Set	C4.5-L	NB	C4.4	NBTree	CITree	CLLTree
Vote	-6.09±3.41	-27.25±13.85	-5.81±4.58	-7.35±5.41	-5.29±3.90	-7.16±4.68
HepatitisS	-171.13±58.85	-180.24±44.34	-160.20±63.97	-77.30±46.28	-72.54±48.90	-34.73±16.08
HepatitisW	-162.52±53.30	-163.69±69.00	-68.05±38.37	-59.09±27.01	-23.69±9.55	-11.58±7.83
average	-55.76	-85.64	-43.93	-31.97	-28.13	-19.74

Table 5. Experimental results for several PETs models: sizes of trees

Data Set	C4.5-L	C4.4	NBTree	CITree	CLLTree
Breast	55	136	8	7	1
Breast(W)	12	31	22	8	3
Chess	35	55	34	26	1
Dermatology	30	33	4	2	1
Heart	19	134	7	1	1
Mushroom	24	24	115	1	1
SPLICE	184	214	50	2	34
Vote	6	13	11	7	1
HepatitisS	64	108	10	8	2
HepatitisW	27	147	13	24	2
average	45.6	89.5	27.4	8.6	4.7

4 Conclusion

In this paper, a comparative study of six well-known probability estimation methods has been presented. Each method has been critically reviewed, and its behavior tested on several datasets. Some objective evaluations of performance to accurate class probability estimation are given by each method.

To sum up, we have shown that:

- Building PETs are greedy and recursive process, similar to building traditional decision trees. In this paper, we conclude a framework for PETs algorithms in Figure 1.
- Our results present an important caveat: although larger trees may not be more accurate, that does not mean that they are not better models. As shown by the results, C4.4 fairly outperforms a traditional decision tree represented by C4.5-L. Moreover, C4.4 also outperforms other PETs presented in this paper in terms of ACC.
- Another significant observation is that PETs (NBTree, CITree and CLLTree), as a combination of a decision tree and a naïve Bayes, can yield more accurate probability estimates than naïve Bayes and traditional decision trees.
- The representation of Joint distribution at leaves, such as CITree and CLLTree, is more reliable and accurate, and outperforms class probability estimation of NBTree.

```
$T$ : a PET
$S$ :  a set of training samples
$A$ : a set of attributes

Begin
for each attribute $A_i \in A$ do
    Partition $S$ into $S_1, S_2, \cdots, S_k$, where $k$ is the number of possible values of attribute $A_i$. Each subset is corresponding to a value of $A_i$.
    Create a naïve Bayes model for each $S_i$.
    Evaluate the split on the attribute $A_i$ in term of score function.

Choose the attribute $A_t$ with the highest split score.
If the split score is not significant better than the score of the current node, create a naïve Bayes classifier for the current node and return.
Else
    for all the values $S_a$ of $A_t$ do
        $PET(T_a, S_a, A - A_t)$
    add $T_a$ as a child of $T$

Return $T$
End;
```

Figure 1. A framework for PETs algorithms

This paper on PETs is manifestly incomplete. Several other PETs methods presented in the literature are neglected by us. Padhraic Smyth et al. [9] proposed a novel method for combining decision trees and kernel density estimators. Dragos and Thomas [26] introduced a B-LOTs algorithm. In addition, many researchers specifically discussed a resemble decision tree induction with bagging. They conclude that bagging is not a good choice if we aim to calibrate class probabilities of decision trees.

Although these PETS presented in this paper can produce accurate probability estimations, they are a greedy process in building trees. In our future research, we will devote ourselves to optimize the algorithms, and reduce computation. In addition, we will also extend PETs to work on data with higher dimensionality.

Acknowledgments

We thank the contributors to and the librarians of the UCI repository for facilitating experimental research in machine learning.

We gratefully acknowledge all the researchers from the Shanghai University of Traditional Chinese Medicine for the TCM databases. This research is partly supported by subprogram of China 973 project (No. 2006CB504801), Important National Science & Technology Specific Projects (No. 2009ZX10004601).

References:

[1] Sreerama K. Murthy, Automatic Construction of Decision Trees from Data: A Multi-disciplinary Survey, *Data Mining and Knowledge Discovery*, Vol.2, 1998, pp.345–389.

[2] F. Provost, V. Kolluri, A Survey of Methods for Scaling up Inductive Algorithms, *Data Mining and Knowledge Discovery*, Vol.3, No.2, 1999, pp.131-169.

[3] J. R. Quinlan, *C4.5: Programs for Machine Learning*, San Mateo, Calif.: Morgan Kaufmann, 1993.

[4] F. Provost, T. Fawcett, R. Kohavi, The Case Against Accuracy Estimation for Comparing Induction Algorithms, *Proceedings of the Fifteenth International Conference on Machine Learning, Morgan Kaufmann,* 1998, pp.445-453.

[5] F. Provost, P. Domingos, Tree Induction for Probability-Based Ranking, *Machine Learning,* Vol.52, No.3, 2003, pp.199-215.

[6] Han Liang, Harry Zhang and Yuhong Yan, Decision Trees for Probability Estimation: An Empirical Study, *Proceedings of the 18th IEEE International Conference on Tools with Artificial Intelligence*, 2006, pp.756-764.

[7] David J. Hand, Robert J. Till, A Simple Generalisation of Area Under the ROC Curve for Multiple Class Classification Problems, Machine Learning, Vol.45, pp.171-186, 2001.

[8] Daniel Grossman, Pedro Domingos, Learning Bayesian Network Classifiers by Maximizing Conditional Likelihood, *Proceedings of the 21st International conference on Machine Learning, Banff, Canada*, 2004, pp.361-368.

[9] Padhraic Smyth, Alexander Gray, Usama M. Fayyad, Retrofitting Decision Tree Classifiers Using Kernel Density Estimation, *in Proceeding Decision Tree Classifiers Using Kernel Density Estimation,* 1995, pp.506-514,

[10] N. Friedman, M. Goldszmidt, Learning Bayesian Networks with Local Structure, *in Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence,* 1996, pp.252-262.

[11] P. Domingos, MetaCost: A General Method for Making Classifiers Cost-Sensitive, *in Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Ming,* 1999, pp. 155-164.

[12] A. Danyluk, F. Provost, *Telecommunications Network Diagnosis*, in Kloesgen, W., & Zytkow J. (Eds.), Handbook of Knowledge Discovery and Data Ming, 2000. To Appear.

[13] F. Provost, P. Domingos, *Well-trained PETs: Improving Probability Estimation Trees*, Technical Report IS-00-04, Stern School of Business, New York University, 2000.

[14] Charles X. Ling, Robert J. Yan, Decision Tree with Better Ranking, *in Proceedings of the Twentieth International Conference on Machine Learning,* 2003, pp.480-487.

[15] Ron Kohavi, Scaling Up the Accuracy of Naïve-Bayes Classifiers: A Decision-Tree Hybrid, *in Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 1996, pp.114-119.

[16] Lior Rokach, Oded Maimon, *Data Mining with Decision Trees: Theory and Applications*, World Scientific Publishing Co. Pte. Ltd., 2008.

[17] Harry Zhang, Jiang Su, Conditional Independence Trees, *European Conference on Machine Learning*, 2004, pp.513-524.

[18] Jiang Su, Harry Zhang, Representing Conditional Independence Using Decision Trees, *American Association for Artificial Intelligence*, 2005, pp.874-879.

[19] Jiang Su, Harry Zhang, Learning Conditional Independence Tree for Ranking, *Fourth IEEE International Conference on Data Mining (ICDM'04),* 2004, pp.531-534.

[20] Harry Zhang, Jiang Su, Learning probabilistic decision trees for AUC, *Pattern Recognition Letters,* Vol.27, 2006, pp.892-899.

[21] Han Liang, Yuhong Yan, Learning Naïve Bayes Tree for Conditional Probability Estimation, *Advances in Artificial Intelligence,* Vol.4013, 2006, pp.455-466.

[22] Han Liang, Yuhong Yan, Harry Zhang, Learning Decision Trees with Log Conditional Likelihood, *International Journal of Pattern Recognition and Artificial Intelligence,* Vol.24, No.1, 2010, pp.117-151.

[23] C. Blake, C. J. Merz, UCI repository of machine learning database.

[24] Usama M. Fayyad, Keki B. Irani, Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, *Proceedings of the International Joint Conference on Uncertainty in AI,* 1993, pp.1022-1027.

[25] Jin Huang, Charles X. Ling, Using AUC and Accuracy in Evaluating Learning Algorithms, *IEEE Transactions on Knowledge and Data Engineering,* Vol.17, No.3, 2005, pp.299-310.

[26] Dragos D. Margineantu, Thomas G. Dietterich, Improved Class Probability Estimates from Decision Tree Models, in Holmes, C. (Ed.), *Nonlinear Estimation and Classification*, The Mathematical Sciences Research Institute, University of California Berkeley, 2001, pp.169-184.