

Manuscript Number: IJPRAI-D-07-00173R2

Title: Learning Decision Trees With Log Conditional Likelihood

Article Type: Research Paper

Keywords: Classification; Log Conditional Likelihood; Probability Estimation Tree; Discriminative Learning; AUC

Corresponding Author: Mr. Han Liang, M.Sc.

Corresponding Author's Institution: University of Alberta

First Author: Han Liang, M.Sc.

Order of Authors: Han Liang, M.Sc.; Yuhong Yan, Ph.D.; Huajie Zhang, Ph.D.

Abstract: In machine learning and data mining, traditional learning models aim at high classification accuracy. However, accurate class probability prediction is more desirable than classification accuracy in many practical applications, such as medical diagnosis. Although it is known that decision trees can be adapted to class probability estimators in a variety of approaches, and the resulting models are uniformly called Probability Estimation Trees (PETs), the performances of these PETs in class probability estimation, have not yet been investigated. We begin our research by empirically studying PETs in terms of class probability estimation, measured by Log Conditional Likelihood (LCL). We also compare a PET called C4.4 with other representative models, including Naive Bayes, Naive Bayes Tree, Bayesian Network, KNN and SVM, in LCL. From our experiments, we draw several valuable conclusions. First, among various tree-based models, C4.4 is the best in yielding precise class probability prediction measured by LCL. We provide an explanation for this and reveal the nature of LCL. Second, compared with non tree-based models, C4.4 also performs best. Finally, LCL does not dominate another well-established relevant metric -- AUC, which suggests that different decision-tree learning models should be used for different objectives. Our experiments are conducted on the basis of 36 UCI sample sets. We run all the models within a machine learning platform - Weka.

We also explore an approach to improve the class probability estimation of Naive Bayes Tree. We propose a greedy and recursive learning algorithm, where at each step, LCL is used as the scoring function to expand the decision tree. The algorithm uses Naive Bayes created at leaves to estimate class probabilities of test samples. The whole tree encodes the posterior class probability in its structure. One benefit of improving class probability estimation is that both classification accuracy and AUC can be possibly scaled up. We call the new model LCL Tree (LCLT). Our experiments on 33 UCI sample sets show that LCLT outperforms all state-of-the-art learning models, such as Naive Bayes Tree, significantly in accurate class probability prediction measured by LCL, as well as in classification accuracy and AUC.

Response to Reviewers: Please see the separate file

Point-to-point Responses to Reviewers' Comments

Reviewer #1: I have three problems for this revised paper.

- 1. There is an obvious cited error in the sixth paragraph in page 12.**
- 2. The resolution of Fig. 1 should be improved. The words in x-axis and y-axis are too blurred.**
- 3. Most importantly, although time complexity is not the focus of this paper, but I do not agree that you just remove the discussion. If an algorithm reaches very good probability prediction but needs terrible execution time, is this algorithm valuable? The answer is absolutely 'no'. In my intuition, I guess LCLT might have a similar time complexity to that of NBT, but I am not very sure. You should give more discussion about the time complexity instead of just ignoring it.**

- 1. There is an obvious cited error in the sixth paragraph in page 12.**

Responses: This error has been properly corrected. Please see the sixth paragraph of Page 12.

- 2. The resolution of Figure 1 should be improved. The words in x-axis and y-axis are too blurred.**

Responses: This error has been properly corrected. Please see Figure 1 of Page 5.

- 3. Most importantly, although time complexity is not the focus of this paper, but I do not agree that you just remove the discussion. If an algorithm reaches very good probability prediction but needs terrible execution time, is this algorithm valuable? The answer is absolutely 'no'. In my intuition, I guess LCLT might have a similar time complexity to that of NBT, but I am not very sure. You should give more discussion about the time complexity instead of just ignoring it.**

Responses: This error has been properly corrected. Please see the first paragraph of Page 20. We have added one paragraph to discuss the time complexity of LCLT. The time complexity of LCLT is equal to NBT.

Reviewer #1: The revised version improves a lot, though I think the authors should not have deleted some subsections.

Responses: The paragraph that discusses the time complexity of LCLT has been added to the paper. Please see the first paragraph of Page 20. The time complexity of LCLT is equal to NBT.

Associate editor: The authors need to add in a discussion of complexity. Also, the errors and resolution issues must be fixed.

Responses: The paragraph that discusses the time complexity of LCLT has been added to the paper. Please see the first paragraph of Page 20. The citation error and the resolution problem have been properly fixed. Please see the sixth paragraph of Page 12 and Figure 1 of Page 5.

International Journal of Pattern Recognition and Artificial Intelligence
© World Scientific Publishing Company

LEARNING DECISION TREES WITH LOG CONDITIONAL LIKELIHOOD

HAN LIANG

*Department of Computing Science
University of Alberta
Edmonton, Alberta T6G 2E1, Canada
hliang2@ualberta.ca*

YUHONG YAN

*Faculty of Computer Science and Software Engineering
Concordia University
Montreal, Quebec H3G 1M8, Canada
yuhong@cse.concordia.ca*

HARRY ZHANG

*Faculty of Computer Science
University of New Brunswick
Fredericton, New Brunswick E3B 5A3, Canada
hzhang@unb.ca*

In machine learning and data mining, traditional learning models aim at high classification accuracy. However, accurate class probability prediction is more desirable than classification accuracy in many practical applications, such as medical diagnosis. Although it is known that decision trees can be adapted to class probability estimators in a variety of approaches, and the resulting models are uniformly called *Probability Estimation Trees* (PETs), the performances of these PETs in class probability estimation, have not yet been investigated. We begin our research by empirically studying PETs in terms of class probability estimation, measured by *Log Conditional Likelihood* (LCL). We also compare a PET called C4.4 with other representative models, including Naïve Bayes, Naïve Bayes Tree, Bayesian Network, KNN and SVM, in LCL. From our experiments, we draw several valuable conclusions. First, among various tree-based models, C4.4 is the best in yielding precise class probability prediction measured by LCL. We provide an explanation for this and reveal the nature of LCL. Second, compared with non tree-based models, C4.4 also performs best. Finally, LCL does not dominate another well-established relevant metric – AUC, which suggests that different decision-tree learning models should be used for different objectives. Our experiments are conducted on the basis of 36 UCI sample sets. We run all the models within a machine learning platform - Weka.

We also explore an approach to improve the class probability estimation of Naïve Bayes Tree. We propose a greedy and recursive learning algorithm, where at each step, LCL is used as the scoring function to expand the decision tree. The algorithm uses Naïve Bayes created at leaves to estimate class probabilities of test samples. The whole tree encodes the posterior class probability in its structure. One benefit of improving class probability estimation is that both classification accuracy and AUC can be possibly

scaled up. We call the new model *LCL Tree* (LCLT). Our experiments on 33 UCI sample sets show that LCLT outperforms all state-of-the-art learning models, such as Naïve Bayes Tree, significantly in accurate class probability prediction measured by LCL, as well as in classification accuracy and AUC.

Keywords: Classification; Log Conditional Likelihood; Probability Estimation Tree; Discriminative Learning; AUC

1. Introduction

Classification is a fundamental problem in machine learning and data mining, in which a learning model is induced from a set of training samples represented by a vector of attribute values and a class label. We denote attribute set $\mathbf{A} = \{A_1, A_2, \dots, A_n\}$ and an assignment of value to each attribute in \mathbf{A} by a corresponding bold-face lower-case letter \mathbf{a} . We use C to denote the class label and c to denote its value. Thus, a sample $\mathbf{s} = (\mathbf{a}, c)$, where $\mathbf{a} = (a_1, a_2, \dots, a_n)$ and a_i is the value of attribute A_i . An inductive learning model is a function h that predicts the assignment of C for a test sample \mathbf{s}_t , i.e. $h(\mathbf{s}_t) = c$. The function h is called a hypothesis.

In a traditional learning scenario, a learning model is evaluated by its classification accuracy. However, accurate class probability prediction generated by learning models is more desirable in real life. For example, in cost-sensitive classification problems, such as medical diagnosis, knowing accurate class probabilities is crucial in making a good decision. Determining only the decision boundary, which is enough for classification, is not satisfiable. Furthermore, accurate class probabilities can help to improve classification accuracy, though it is not a necessary condition. At last, good class probability estimation can possibly result in a precise class probability-based ranking list of test samples.

Various inductive learning models can be categorized into two approaches: the probability-based approach and the decision boundary-based approach. In a probability-based learning model, a joint class probability distribution $\hat{p}(\mathbf{s}_t, c)$ is learned from training samples as a hypothesis. \mathbf{s}_t is classified into class c with the maximum posterior class probability $\hat{p}(c|\mathbf{s}_t)$ (or simply class probability), as shown below.

$$h(\mathbf{s}_t) = \arg \max_{c \in C} \hat{p}(\mathbf{s}_t, c) = \arg \max_{c \in C} \hat{p}(c|\mathbf{s}_t) \hat{p}(\mathbf{s}_t), \quad (1)$$

where \mathbf{s}_t is classified into class c with the maximum of $\hat{p}(c|\mathbf{s}_t)$.

Decision trees are well-known as decision boundary-based models. However, they can be easily adapted to class probability estimators by use of class distributions at leaves. Unfortunately, their class probability estimation has been found inefficient⁸. Numerous techniques have been proposed to improve traditional decision trees on class probability estimation, of which *Laplace* correction, turning off pruning, *m*-Branch, the confusion factor algorithm and ensemble learning have been adopted. In this paper, we conduct a systematic experimental study on their efficacy for class probability estimation, based on 36 UCI² sample sets. We use a performance metric,

called *Log Conditional Likelihood* (LCL), to evaluate and compare learning models. Our paper draws a series of interesting conclusions. Among various tree-based models, C4.4 is the best in yielding accurate class probability prediction measured by LCL. C4.4 also performs best compared with non tree-based models. LCL is not experimentally consistent with another class probability-based performance metric, called *Area Under the ROC Curve* (AUC) ¹⁴.

Furthermore, although our experimental results indicate that Naïve Bayes Tree outperforms most of the models in classification accuracy and AUC, it does not work well in LCL and is just competitive to Naïve Bayes. We aim to explore an approach to improve the class probability estimation of Naïve Bayes Tree. Similar to the growth process of Naïve Bayes Tree, our proposed learning algorithm is a greedy and recursive procedure. In each step of expanding the tree, LCL is used as the scoring function to select the best attribute to split. The splitting process ends when no candidate attribute can improve the function or there are less than 30 training samples at current node. Then, at the deepest nodes, Naïve Bayes models are created. The resulting model optimizes the estimation of class probability. We call the new model *LCL Tree* (LCLT). On a large suite of benchmark sample sets, LCLT significantly outperforms all of the state-of-art learning models in terms of classification accuracy, AUC and LCL.

The paper is organized as follows. Section 2 presents the research background in LCL, its relations to two class probability-based metrics: classification accuracy and AUC, and finally summarizes our research motivations. Section 3 reviews the existing work on augmenting decision trees for better class probability-based ranking measured by AUC. In Section 4, we empirically study decision trees in terms of LCL. The experimental configuration and methodology are described and empirical results are analyzed. Section 5 presents LCLT and its experimental results. We draw our conclusions and outline future work in Section 6.

2. Log Conditional Likelihood and Motivations of Our Research

2.1. Performance Metrics vs. Class Probability Estimation

In this subsection, we analyze the performance metrics for different learning tasks and their relations to class probability estimation.

2.1.1. Classification Accuracy

Classification is one of the fundamental issues in machine learning and data mining. In classification scenarios, the goal is to learn a model from training samples, which correctly assigns class labels to test samples. The performance of a learning model is usually measured by its classification accuracy (ACC). ACC is calculated as the percentage of correctly classified testing samples over all test samples, as Eq. (2) describes.

$$ACC = \frac{1}{N} \sum I \{h(\mathbf{s}_t) = c_{true}\}, \quad (2)$$

where N is the number of test samples and c_{true} is the true class label \mathbf{s}_t belongs to. The indicator function $I\{x=y\}$ is one if $x=y$ and zero otherwise.

Classification can be performed by decision boundary-based models, such as decision trees, and class probabilities of test samples can be used to determine class boundaries. For example, assume that classification threshold is 0.5. If the class probability estimation $\hat{p}(+|\mathbf{s}_+)$ is larger than 0.5, $+$ is assigned to \mathbf{s}_+ . The class boundary at the leaf, where \mathbf{s}_+ falls, will be shifted towards the negative class. Better class probability estimation can possibly improve classification accuracy. However, it is not guaranteed. For another example, if a learning model improves $\hat{p}(+|\mathbf{s}_+) = 0.3$ to $\hat{p}(+|\mathbf{s}_+) = 0.4$, this model still gives an incorrect result, and its ACC is not improved.

2.1.2. Cost-sensitive Learning

In some practical applications, we need to assign different costs to different types of misclassification. For instance, in medical diagnosis, the cost incurred by predicting a patient who has lung cancer as the one not to have, is significantly greater than the converse. The optimal decision for a test sample \mathbf{s}_t is to assign \mathbf{s}_t to class c_i that minimizes the conditional risk ⁷, defined in Eq. (3).

$$h(\mathbf{s}_t) = \arg \min_{c_i \in C} \sum_{c_j \in C - c_i} \hat{p}(c_j|\mathbf{s}_t)C(c_i, c_j), \quad (3)$$

$C(c_i, c_j)$ is the cost of classifying \mathbf{s}_t into class c_i , while its true class is c_j . $\hat{p}(c_j|\mathbf{s}_t)$ is the class probability by which \mathbf{s}_t belongs to c_j . Apparently, we need a good estimate of the class probability $\hat{p}(c_j|\mathbf{s}_t)$. Therefore, accurate class probability estimation is required for cost-sensitive learning.

2.1.3. Class Probability-based Ranking

The ROC curve has been introduced to machine learning research in recent years, in response to classification tasks with varying class distributions or misclassification costs ²⁸. For a binary-class problem, we assume that threshold t is the probability of randomly chosen negative points belonging to the positive class. Therefore, a ROC curve is constructed by plotting different points (FP,TP) as we move threshold t between the extreme points -0 and 1 ¹⁶. $TP(t)$ is the probability that a randomly chosen positive point has a higher probability of belonging to the positive class than t . $FP(t)$ is the probability that a randomly chosen negative point has a higher probability of belonging to the positive class than t . Thus, we can obtain a ROC curve of a learning model by moving the threshold t to cover the whole FP distribution. Fig. (1) ⁴ shows a graph of four ROC curves, each of which represents a learning model, from **A** to **D**. A ROC curve **X** is said to dominate another ROC curve **Y** if **X** is always above **Y**. This means that the learning model of **X** always has a lower expected cost than that of **Y**, over all possible class distributions and misclassification costs. In this graph, **A** and **B** dominate **D**. However, there is usually no clear

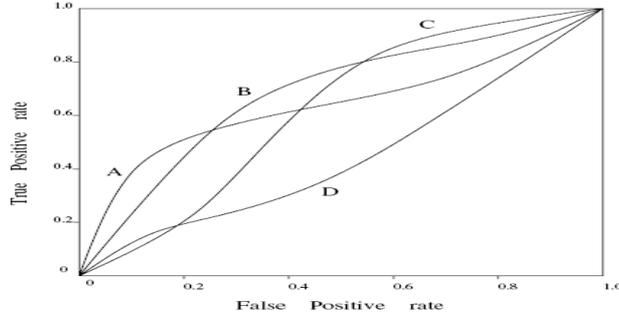


Fig. 1. The ROC Curves

dominating relation between two ROC curves. For instance, curve **A** and **B** are not dominating each other in the whole range. The *Area Under the ROC Curve*, or simply AUC, has been used to provide a good “summary” for the performances of ROC curves. A simple approach to calculate the AUC value for a binary-class sample set is shown below ¹⁴.

$$AUC = \frac{S_+ - n_+(n_+ + 1)/2}{n_+n_-}, \quad (4)$$

where n_+ or n_- is respectively the number of positive or negative samples, and $S_+ = \sum_{|+|} r_i$, where r_i is the i th positive sample in the ranking list. In an extreme situation, the AUC value of a model could reach one if any positive sample is ranked behind all negative samples. For multi-class situations, we separately calculate the AUC value for each pair of class memberships and average all the values ¹⁴.

Class probability-based ranking sorts a set of test samples according to their probabilities of belonging to a class membership. If their relative orders are correct, their ranks will be correct. For example, assume that \mathbf{s}_+ or \mathbf{s}_- respectively denotes a positive or a negative sample, and their actual class probabilities are $p(+|\mathbf{s}_+) = 0.9$ and $p(-|\mathbf{s}_-) = 0.8$. A learning model, which gives class probability estimates $\hat{p}(+|\mathbf{s}_+) = 0.55$ and $\hat{p}(-|\mathbf{s}_-) = 0.54$, will generate a correct order of \mathbf{s}_+ and \mathbf{s}_- in the positive class ranking list, regardless of the errors in class probability estimation. From this instance, we can learn that improving class probability estimation can potentially improve AUC.

2.2. Generative Learning vs. Discriminative Learning

If our target is to obtain accurate class probability estimation, could the existing learning models generate good class probabilities? Unluckily, most of traditional learning models are designed as *generative* models in the sense that they are con-

structed by maximizing joint class probability estimation, as shown in Eq. (5).

$$LL(\mathbf{\Gamma}|\mathbf{S}) = \sum_{i=1}^M \log \hat{p}(a_1^i, a_2^i, \dots, a_n^i, c^i), \quad (5)$$

where $\log \hat{p}(a_1^i, a_2^i, \dots, a_n^i, c^i)$ is the log value of joint class probability produced by learning model $\mathbf{\Gamma}$. M is the number of test samples in sample set \mathbf{S} , c^i denotes the class label of the i_{th} test sample and $\{a_1^i, a_2^i, \dots, a_n^i\}$ denotes the set of attribute values of that sample. *Generative learning* is a direct approach to marginalize over joint class probabilities in order to make inferences or predictions. However, this is inefficient because, for classification tasks, there is no need to characterize joint class probabilities. Furthermore, if we learn an optimal model by use of generative learning, a huge number of training samples are required, which is always impossible in practical applications. At last, since generative learning is associated with joint class probabilities, the distinction between the roles of variables (attributes and the class label) will not be detected.

For classification purposes, *Log Conditional Likelihood* (LCL) is relevant, where

$$LCL(\mathbf{\Gamma}|\mathbf{S}) = \sum_{i=1}^M \log \hat{p}(c^i | a_1^i, a_2^i, \dots, a_n^i). \quad (6)$$

Note that,

$$LL(\mathbf{\Gamma}|\mathbf{S}) = LCL(\mathbf{\Gamma}|\mathbf{S}) + \sum_{i=1}^M \log \hat{p}(a_1^i, a_2^i, \dots, a_n^i). \quad (7)$$

Maximizing LL can under-perform learning models, since the contribution of LCL is always balanced by the more general term $\sum_{i=1}^M \log \hat{p}(a_1^i, a_2^i, \dots, a_n^i)$ ²². The solution will be the direct use of LCL as the objective function. Eq. (6) shows that optimizing LCL can result in a model that best approximates the posterior probability of c^i given the attribute set $\{a_1^i, a_2^i, \dots, a_n^i\}$ and further reduces its classification errors^{22,21}. This is a form of *discriminative learning*, which intentionally focuses on extending the margins between class memberships. In contrast to generative models, discriminative models directly build posterior class probabilities using discriminant functions. This is indeed the goal of the Support Vector Machine (SVM) approach to classification⁵. SVM directly maximizes the margin of a linear separator between two sets of points in an Euclidean space.

2.3. Motivations for Our Research

It has been observed that traditional decision trees can only generate piecewise constant estimates of class probabilities, as a result, all the test samples classified by a tree leaf share the same class probabilities. Moreover, the class probability estimates yielded at leaves are highly biased. According to these inherit constraints, a lot of work have been done on improving the class probability estimation of

decision trees and the resulting models are collectively referred to as *Probability Estimation Trees* (PETs).

However, most methods mentioned in Section 3.1 are intended to improve class probability-based ranking measured by AUC. AUC is a relative evaluation metric, that is, the correctness of ranking depends on the relative position of a sample among a set of the others. However, LCL is directly calculated by adding up log values of class probabilities generated by a learning model (see Eq. (6)). Therefore, LCL and AUC represent two aspects of class probability estimation: *reliability* and *resolution*. Dawid⁶ described these two conceptual criteria when studying how effective class probability predictions are. *Reliability* indicates that class probabilities should be reliable and accurate. In other words, when we assign a positive class probability $\hat{p}(+|\mathbf{s}_t)$ to a test sample \mathbf{s}_t , there should be roughly $1 - \hat{p}(+|\mathbf{s}_t)$ of the class probability for that sample not occurring. *Resolution* presents that test samples should be easily ranked in terms of their class probabilities. Thus, LCL can be used to evaluate the reliability of class probability estimation and AUC can be used to evaluate its resolution performance.

In recent years, given a set of samples with known true class labels, LCL has been used as a performance metric^{22,10,11}. In this paper, we use LCL as a performance metric to evaluate the class probability estimation produced by PETs. We conduct an empirical study to answer a series of relevant questions, such as 1) if LCL is a good metric for accurate class probability estimation; 2) with the objective of yielding accurate class probabilities, if LCL can be used to induce a learning model; 3) if LCL connects to classification accuracy and AUC. We will use ACC, AUC and LCL to evaluate learning models in our experiments.

3. Probabilistic Models

3.1. Probability Estimation Trees

In the decision boundary-based theory, a test sample \mathbf{s}_t is categorized into class c_j if \mathbf{s}_t falls into the decision area corresponding to c_j . A decision tree, such as ID3²⁶ and C4.5²⁷, can be transformed into a class probability estimator using raw class frequencies at its leaves. For instance, if a leaf has a set of class frequencies n_1, n_2, \dots, n_k , the estimated probability of each class membership at this leaf is represented as $\hat{p}(c_i|\mathbf{s}_t) = n_i / \sum_{j=1}^k n_j$. But this method is poor for two reasons⁸:

- (1) the estimated class probabilities are systematically shifted toward zero or one.
- (2) test samples will be assigned the same class probabilities if they fall into the same leaf.

Probability Estimation Tree (PET) is a tree that estimates the probability of class membership²⁵. PET is seeing increasing use in a variety of ways, such as ranking test samples according to their class probabilities²⁵ or specifying conditional probabilities in Bayesian models⁹. Several approaches of learning PET have been proposed in the literature. With the goal of better class probability-based ranking,

Provost and Domingos²⁵ introduced two methods to improve class probability estimation of a decision tree. First, by use of *Laplace* correction at the leaves, class probabilities can be systematically smoothed towards the priori class distributions. Second, by removing pruning and collapsing postprocessing steps in C4.5, a decision tree can keep some branches that are crucial for accurate class probability prediction. The new model is called C4.4. However, C4.4 still has two contradictions:

- (1) turning off pruning may result in a fairly large tree. Therefore, within some leaves having only few training samples, the class probability estimation is unreliable.
- (2) a large tree always overfits training sample sets, and the yielded class probabilities could be still doubtful even *Laplace* correction has been applied.

In addition, values of class probabilities can easily repeat, which may substantially decrease the quality of ranking test samples based on their class probabilities.

Some researchers have noticed that the information used to estimate the class probabilities for a test sample should not be limited to the leaf where the sample falls. Ling and Yan²⁰ proposed a model called *Confusion Factor Tree* (CFT), which assigns a fixed confusion factor f to each internal node of a decision tree. This parameter measures the probability of errors if altering the value of a tested attribute at a decision node due to noise introduced in data collection. Therefore, CFT produces the class probabilities of a test sample \mathbf{s}_t with reference to the class probabilities from all leaves. The contribution of each leaf is determined by the number of its unequal path attribute values compared to \mathbf{s}_t . The class probability of \mathbf{s}_t is estimated by the weighted average of the contributions from all leaves, as Eq. (8) depicts.

$$\hat{p}(c_i|\mathbf{s}_t) = \frac{f^q \hat{p}_{L_i}}{\sum f^q}, \quad (8)$$

where \hat{p}_{L_i} is the generated class probability at leaf L_i and q is the number of unequal attribute values.

Ferri *et al.*³ introduced a smoothing method, called m -Branch. M -Branch is a recursive, root-to-leaf extension of m probability estimation. At each leaf, a class probability is generated by propagating the class probabilities of each of the leaf's parent nodes from the root down to itself. Eq. (9) is the formal expression of the m -Branch method:

$$\hat{p}_{child}(c_i|\mathbf{s}_t) = \frac{n_i + m\hat{p}_{parent}(c_i|\mathbf{s}_t)}{\sum_{j=1}^k n_j + m}, \quad (9)$$

where $\hat{p}_{parent}(c_i|\mathbf{s}_t)$ is the class probability smoothed from the root to its direct parent node, k represents the number of class values, and parameter m is adjusted by the depth and cardinality (the number of training samples associated with a node) of the current node. Moreover, the authors also introduced a new tree splitting criterion that chooses the split with local highest AUC value, rather than entropy-based criteria, such as information gain. Experiments showed that this splitting

criterion results in trees with equal or better AUC performances without sacrificing classification accuracy, compared with traditional decision trees.

Bootstrap aggregating (bagging), an ensemble approach that aggregates class probabilities from a suite of base learners. Bagging is applied on decision trees to overcome the instability of class probability estimation. This method creates an ensemble of trees from the original training sample set. Each tree is generated by a bootstrap duplicate of the original set. The final output is formed by use of a plurality vote among these trees. Recently, bagging has been applied to improve class probability-based ranking of test samples¹. One drawback of bagging is that its outputs are not easily comprehensible.

3.2. Bayesian Networks

A Bayesian Network (BN)²⁴ consists of a directed acyclic graph \mathbf{G} that encodes conditional independence among a set of attribute nodes and a class node, and a set \mathbf{P} that represents local distributions of nodes. A local distribution is typically specified by a *Conditional Probability Table* (CPT). Each attribute node is independent of its non-descendants in the graph given the state of its parents. What a BN represents is a joint class distribution of all nodes $A_1, A_2 \dots A_n, C$. Eq. (10) gives the formal expression

$$\hat{p}(A_1, A_2 \dots A_n, C) = \prod_i \hat{p}(A_i | \text{parent}(A_i)) \hat{p}(C), \quad (10)$$

where $\text{parent}(A_i)$ is the set of the i_{th} attribute's parents, which may include the class label C and other attributes.

In recent years, BNs have been widely used in classification problems, such as pattern recognition and fault diagnosis. Naïve Bayes (NB) is the simplest BN model in that each attribute node has only the class node as its parent. NB assumes that there is no attribute dependency given the class node. Thus, NB is easy to be built up. However, unfortunately, this strong independence assumption is unrealistic in real world. Correlations among attributes in a given domain are common. For example, in medical diagnosis, certain symptoms are more common among older patients than younger ones, regardless of whether they are ill. Such correlations introduce dependencies into the probabilistic summaries that can degrade NB's classification accuracy.

Tree-Augmented Naïve Bayes (TAN), proposed by Friedman *et al.*²², approximates the interaction between attributes using a tree structure imposed on the NB framework. In TAN, an attribute node can have at most one parent attribute other than the class node, and these attribute-attribute arcs form a tree. TAN is a good trade-off model that balances the quality of the approximation of correlations among attributes and the computational complexity of the learning process.

3.3. Naïve Bayes Tree

Another alternative approach to improve decision trees is to stop splitting at a certain level and put a probability density estimator at each leaf. Kohavi¹⁷ proposed a *Naïve Bayes Tree* (NBT) that uses a decision tree as the general structure and deploys Naïve Bayes models at the leaves. This hybrid model first uses classification accuracy of local Naïve Bayes models as the scoring function to do univariate splits, and when splitting does not increase the accuracy, a leaf Naïve Bayes model is created at current node. In NBT, the attributes of a training sample are grouped into two sets: $\mathbf{A} = \mathbf{A}_p \cup \mathbf{A}_l$, where \mathbf{A}_p is the set of path attributes and \mathbf{A}_l is the set of leaf attributes. The paper showed the significant improvement of NBT on classification accuracy compared to C4.5 and Naïve Bayes, but it did not mention its performance on class probability estimation. Su and Zhang³² proposed one encoding of $\hat{p}(\mathbf{A}, C)$ in the diagram of NBT. The proposed *Conditional Independence Tree* (CIT) denotes $\hat{p}(\mathbf{A}, C)$ as

$$\hat{p}(\mathbf{A}, C) = \alpha \hat{p}(C|\mathbf{A}_p(L)) \hat{p}(\mathbf{A}_l(L)|\mathbf{A}_p(L), C), \quad (11)$$

where α is a normalization factor. The term $\hat{p}(C|\mathbf{A}_p(L))$ denotes the probability of a class membership given the joint distribution of path attributes. The term $\hat{p}(\mathbf{A}_l(L)|\mathbf{A}_p(L), C)$ is generated from the leaf attributes presented by Naïve Bayes. From the conditional independence assumption of Naïve Bayes, Eq. (12) stands

$$\hat{p}(\mathbf{A}_l|\mathbf{A}_p(L), C) = \prod_{i=1}^n \hat{p}(A_{l_i}|\mathbf{A}_p(L), C), \quad (12)$$

where A_{l_i} represents a leaf attribute. CIT explicitly defines conditional dependence among the path attributes and conditional independence among the leaf attributes. The local conditional independence assumption of CIT is a relaxation of the conditional independence assumption of Naïve Bayes. Further study²⁹ revealed that the local conditional independence explains the “replication” problem. One CIT can be decomposed into a set of trees in order to eliminate the replicated subtrees. The complexity of the sum of these trees is not greater than the original one.

4. Empirical Study, Analysis and Discussion

4.1. Learning Model Introduction

The tree-based learning models used in our experiments are listed below.

C4.5: a learning model that implements a traditional decision tree inductive algorithm.

C4.5-L: C4.5 with *Laplace* correction at its leaves.

C4.5-M: C4.5 with the *m*-Branch method applied on its structure.

CFT4.5: C4.5 with the confusion factor algorithm applied on its structure.

From our preliminary experiments, we learned that using 0.4 as the confusion factor was slightly better than the proposed optimal value 0.3. Therefore, we selected a

new confusion factor in our experiments. Notice that in our previous paper ¹⁹, we assigned 0.5 to the confusion factor, and there is not much difference in experimental results when we change the value of the confusion factor from 0.3 to 0.5.

C4.5-L&B: bagging with C4.5 as base learner. We used *Laplace* correction at the leaves of each C4.5. The ensemble consists of 10 base learners.

C4.4: an improved decision tree for better class probability estimation.

C4.4-M: C4.4 with *m*-Branch method applied on its structure.

CFT4.4: C4.4 with the confusion factor algorithm applied on its structure. We assigned 0.4 to the confusion factor.

C4.4-B: bagging with C4.4 as base learner. The ensemble consists of 10 base learners.

In addition, we also performed another group of experiments on C4.4 and other representative learning models, which are:

NBT: a hybrid model of decision tree and Naïve Bayes.

NB: Naïve Bayes.

TAN: an extended tree-like Naïve Bayes, in which the class node directly points to all attribute nodes and each attribute node can at most have two parents: the class node and another attribute node. The *ChowLiu* algorithm ²² was used to learn the structure in our experiments.

KNN-5: a lazy model that, given a test sample \mathbf{s}_t , finds k nearest training samples as its neighbors. KNN estimates class probabilities of \mathbf{s}_t by a simple vote among the class labels of its neighbors. Eq. (13) is a formal expression of this process.

$$\hat{p}(c_j|\mathbf{s}_t) = \frac{1 + \sum_{i=1}^n I\{c_i = c_j\}w_i}{|C| + \sum_{i=1}^n w_i}, \quad (13)$$

where c_i is the class label of the i th neighbor. The indicator function $I\{x = y\}$ is one if $x = y$, otherwise zero. w_i is the weight of the neighbor (one as default) and $|C|$ represents the number of class labels. We assigned $k = 5$ in our experiments.

SVM: *Sequential Minimal Optimization* (SMO) algorithm was used to train a SVM model. We used logistic regression models to improve outputted class probabilities. In multi-class situations, we chose the method introduced by Wu *et al.* ³⁰, to produce class probabilities for SVM.

4.2. Samples Sets

We used 36 UCI ² sample sets to conduct the experiments within Weka ³¹, a machine learning platform. Tab. (1) is a brief description of these sample sets. All sample sets came from UCI repository. We adopted four steps to preprocess these sample sets:

- (1) We used the *ReplaceMissingValues* filter in Weka to replace all missing values for nominal and numeric attributes with the modes and means from each sample set.

Table 1. Description of sample sets used in our experiments. We downloaded these sample sets in the format of *arff* from Weka web site

Data Set	Size	Attribute	Classes	Missing	Numeric
anneal	898	39	6	Y	Y
anneal.ORIG	898	39	6	Y	Y
audiology	226	70	24	Y	N
autos	205	26	7	Y	Y
balance	625	5	3	N	Y
breast	286	10	2	Y	N
breast-w	699	10	2	Y	N
colic	368	23	2	Y	Y
colic.ORIG	368	28	2	Y	Y
credit-a	690	16	2	Y	Y
credit-g	1000	21	2	N	Y
diabetes	768	9	2	N	Y
glass	214	10	7	N	Y
heart-c	303	14	5	Y	Y
heart-h	294	14	5	Y	Y
heart-s	270	14	2	N	Y
hepatitis	155	20	2	Y	Y
hypoth.	3772	30	4	Y	Y
ionosphere	351	35	2	N	Y
iris	150	5	3	N	Y
kr-vs-kp	3196	37	2	N	N
labor	57	17	2	Y	Y
letter-2000	2000	17	26	N	Y
lymph	148	19	4	N	Y
mushroom	8124	23	2	Y	N
p.-tumor	339	18	21	Y	N
segment	2310	20	7	N	Y
sick	3772	30	2	Y	Y
sonar	208	61	2	N	Y
soybean	683	36	19	Y	N
splice	3190	62	3	N	N
vehicle	846	19	4	N	Y
vote	435	17	2	Y	N
vowel	990	14	11	N	Y
waveform	5000	41	3	N	Y
zoo	101	18	7	N	Y

- (2) Because some learning models used in our experiments, such as NB, can not handle numeric attributes, we determined to use the *Discretize* filter, the unsupervised ten-bin discretization in Weka, to discretize numeric attributes. Then, all the attributes were treated as nominal. Although it has been found that a simple discretization method will worsen the accuracy of a learning model¹⁸, we aim to compare different learning models and reveal the best one measured by a performance criterion, by use of the same experimental methodology.
- (3) Note that, if the number of values of an attribute is almost equal to the number of samples in a sample set, this attribute does not contribute any information for the purpose of prediction, thus we used the *Remove* filter in Weka to delete this type of attributes.
- (4) Due to the high time complexity of KNN and SVM, we used the filter of unsupervised *Resample* in Weka to select samples from *Letter* and generated a new sample set named *Letter-2000*. The selection rate is 10%.

4.3. Experimental Organization

We conducted two groups of experiments. We first systematically studied the performances of tree-based models in producing class probability estimation evaluated by LCL and AUC. In this group, C4.5-based and C4.4-based variants (C4.4, C4.5-L, C4.5-M, CFT4.5, C4.5-L&B, C4.4-M, CFT4.4 and C4.4-B) were included. Then, we empirically investigated the efficacy of other representative learning models on LCL and AUC. C4.4, NBT, NB, TAN, KNN-5 and SVM were considered in this group. Besides, we also analyzed the performances of these classic models provided that the size of sample set is large. In all experiments, the outputs of C4.4 were used as the baseline for comparison.

We implemented LCL, AUC, m -Branch, the confusion factor algorithm within Weka and used the current versions of other learning models and bagging in Weka. For each learning model, we ran ten-fold cross validation ten times to test its performance evaluated by an evaluation metric. Runs with the various learning models were carried out on the same training sample sets and evaluated on the same test sample sets. Finally, we conducted a corrected pairwise two-tailed t -test²³ with a 95% confidence level to compare each pair of learning models. That is, we speak of two empirical results as being “significantly different” only if the difference is statistically significant at the 0.05 level according to the t -test. In all t -test tables, each entry $w/t/l$ means that the model in a row wins in w sample sets, ties in t sample sets, and fails in l sample sets, in contrast with the model in the corresponding column.

In our experiments, *Laplace* correction was used in one of the following forms. Given n_{c_j} training samples that have the class label as c_j , t total training samples and k class labels in a sample set, the frequency-based approach of estimating a class probability is $\hat{p}(c_j) = \frac{n_{c_j}}{t}$, but *Laplace* correction calibrates this by $\hat{p}(c_j) = \frac{n_{c_j} + 1}{t + k}$. Furthermore, in *Laplace* correction, $\hat{p}(a_i|c_j)$ is smoothed by $\hat{p}(a_i|c_j) = \frac{n_{ic_j} + 1}{n_{c_j} + v_i}$, where v_i is the number of values of attribute A_i and n_{ic_j} is the number of training samples in class c_j with $A_i = a_i$.

4.4. Result Analysis and Discussion

As shown in Tab. (2) and Tab. (6), C4.4 performs best among all PETs when class probability estimation measured by LCL is desired. Our observations are summarized below.

- (1) C4.4 fairly outperforms a traditional decision tree represented by C4.5-L, in terms of LCL. t -test results (see Tab. (3)) show that C4.4 wins in 10 sample sets and fails in 5 sample sets, compared with C4.5-L. Notice that C4.4 turns off pruning and collapsing postprocessing steps of C4.5-L. Therefore, we conclude that tree pruning and collapsing could significantly affect the performances of class probability estimation, measured by LCL.
- (2) Compared with an ensemble learning approach – bagging, C4.4 is significantly

better than an ensemble of decision trees in LCL. t -test results (see Tab. (3) and Tab. (7)) show that C4.4 is better than C4.5-L&B in 16 sample sets and better than C4.4-B in 20 sample sets. Bagging estimates the class probabilities of a test sample by use of a simple vote among a group of base learners. It has been verified that bagging is useful in improving class probability-based ranking performances of decision trees, evaluated by AUC ²⁵. However, according to our experimental results, bagging is not a good option if we aim to calibrate class probabilities of decision trees.

- (3) Compared with other methods, such as m -Branch and the confusion factor algorithm, C4.4 substantially outperforms these tree variants in LCL. C4.4 wins C4.5-M in 15 sample sets and wins C4.4-M in 13 sample sets. In addition, C4.4 is significantly better than CFT4.5 in 34 sample sets and better than CFT4.4 in 33 sample sets. Thus, we learn that neither m -Branch nor the confusion factor algorithm can augment decision trees for accurate class probabilities.

We also have the t -tests of AUC experimental results on all PETs (see Tab. (4) and Tab. (8)). There are two valuable observations.

- (1) Although C4.4 is the best model in terms of LCL, all the variants of C4.5 and C4.4 proposed to improve class probability-based ranking quality, are better than C4.4, evaluated by AUC (see Tab. (5) and Tab. (9)). This repeats the experimental results reported by other publications.
- (2) Among all PETs, C4.4-B achieves the best results in AUC, which means bagging is an effective technique in improving class probability-based ranking quality.

Now we reconsider the definition of LCL in Eq. (6). In our experiments, we used real world sample sets for our empirical study and we have no idea about their true class distributions. Eq. (6) indicates if a model overwhelmingly yields higher estimation of $\hat{p}(c_j|\mathbf{s}_t)$ than another, its LCL value can be larger than the other. Therefore, LCL favors a model that gives high class probability estimation no matter what the true class classification is. Furthermore, true class probabilities do not even appear in the definition of LCL. Indeed, when using LCL, we implicitly assume that test sample \mathbf{s}_t in class c_j has the class probability $\hat{p}(c_j|\mathbf{s}_t) = 1$ and $\hat{p}(\neg c_j|\mathbf{s}_t) = 0$. This can explain why C4.4 has better LCL performances than C4.4-B: since C4.4 turns off pruning and collapsing postprocessing steps, most of its leaves have few samples and extremely high class probabilities can occur when frequency-based estimation approach (or even *Laplace* correction) is used at its leaves. In addition, bagging and other smoothing methods are designed to avoid high variance of class probability estimation. That means, given a test sample \mathbf{s}_t , these methods consider more training samples when the leave where \mathbf{s}_t falls has few samples. In summary, LCL is an indirect evaluation metric to true class probabilities. From our experiments, we can also conclude that neither LCL nor AUC dominates the other.

In Tab. (10) and Tab. (11), C4.4 works best among classic models in LCL.

- (1) C4.4 tremendously outperforms Bayesian models in terms of LCL. Compared with NB, *t*-test results (see Tab. (11)) demonstrate that C4.4 wins in 17 sample sets. As a tree-like extension of NB, TAN is also worse than C4.4 in 10 sample sets. Besides, *t*-test results indicate that extending the structure of NB to explicitly represent attribute dependencies is a good way to improve class probability estimation, measured by LCL. TAN achieves better performances than NB in 16 sample sets.
- (2) C4.4 also greatly outperforms KNN, a representative lazy model, in terms of LCL. C4.4 is better than KNN with $k=5$ in 12 sample sets. We also conducted a bench of experiments to compare C4.4 to a series of KNNs with different k (10, 30 and 50) in LCL. Experimental results suggest that the larger k becomes, the worse KNN performs in LCL.
- (3) C4.4 performs better than NBT in LCL. It has been proven that NBT has good performances in ACC and AUC^{13,12}, however, the *t*-test results (see Tab. (11)) show that NBT is inferior in LCL to other learning models and it is just competitive with NB. In Section 5, we propose a new learning algorithm to improve NBT in class probability estimation, where LCL is directly used to conduct the tree growth.
- (4) C4.4 fairly outperforms SVM in terms of LCL. Compared with SVM, C4.4 wins in 12 sample sets and fails in 7 sample sets. However, SVM is still better than most of the models and is competitive with TAN.

Besides, we collected AUC experimental results of classic models (see Tab. (12)) as well. Unfortunately, C4.4 works worst among these models.

- (1) All models significantly outperform C4.4 in terms of AUC. Compared with NBT, C4.4 fails in 20 sample sets and wins only in 2 sample sets. Also, NB works better than C4.4 in 21 sample sets. This is because that C4.4 produces a large amount of repeated class probabilities at leaves, which greatly degrades its class probability-based ranking quality.
- (2) As an extension of NB, TAN is better than NB in 11 sample sets and fails in 4 sample sets. That indicates representing dependence relations among attributes is a good way of improving NB in AUC. In addition, using NB as a local class probability estimator, NBT is superior to NB in 7 sample sets. Therefore, we include that deploying a kernel density estimator is very efficient in class probability-based ranking.
- (3) SVM works as well as TAN, and both of which have better performances than other models on AUC. In Tab. (13), SVM wins in 6 sample sets and fails in 7 sample sets, compared with TAN. And both SVM and TAN have the best performances among all of these classic models.

We are also interested in analyzing LCL performances of these classic learning models on relatively large UCI sample sets, which simulate real sample sets we collect from daily life. We chose ten sample sets on the condition that the num-

ber of samples in each set is above 900. In Fig. (2), *No1* to *No10* respectively denotes *German-credit*, *Hypothyroid*, *Kr-vs-kp*, *Letter-2000*, *Mushroom*, *Segment*, *Sick*, *Splice*, *Vowel*, *Waveform-5000*. Experimental results on these sample sets in the form of absolute LCL values are presented in Fig. (2). From the plot, we learn

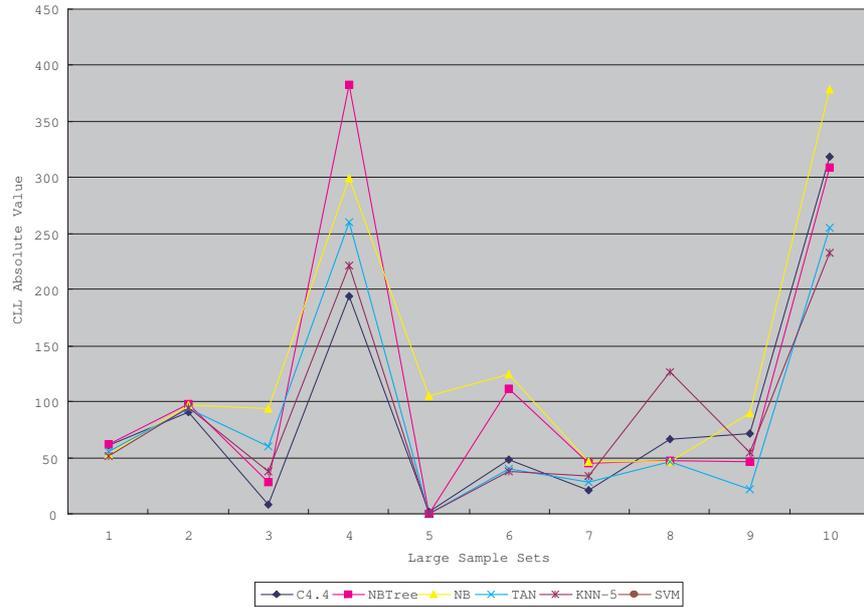


Fig. 2. LCL Performance Curves on Ten Relatively Large Sample Sets

that C4.4 still works best among others (the lowest learning curve). Moreover, TAN and SVM also work well and they can be considered in the applications, where both accurate class probability prediction and high class probability-based ranking quality are required.

5. Learning Naïve Bayes Tree for Better Class Probability Estimation

Naïve Bayes Tree (NBT), proposed by Kohavi ¹⁷, uses a classification accuracy-based splitting procedure to learn a tree structure and deploys Naïve Bayes models at its leaves. The rationale behind this model is that Naïve Bayes works well given a small sample set. Previous experiments have shown that NBT works competitively in terms of classification accuracy and class probability-based ranking ^{13,12}. However, its performance in class probability estimation is undesirable in our experiments. Recently, Zhang and Su ³² explained NBT in a probabilistic approach:

if a probability density estimator which incorporates leaf attributes is put into a tree leaf, the resulting tree encodes accurate class probabilities, together with the conditional probabilities of path attributes. Based on this principle, we propose a new learning algorithm to optimize a tree structure to explore local conditional independencies among attributes, so that a simple leaf probability estimator - Naïve Bayes - can be used to yield accurate class probabilities for test samples. The learning algorithm is a greedy and recursive process where in each iteration LCL is used as the splitting criterion to expand a tree. The algorithm uses Naïve Bayes to estimate the conditional probabilities of leaf attributes given the class label and the path attributes. We name the resulting model *LCL Tree* (LCLT). The whole tree encodes class distributions by its optimized tree structure.

5.1. Tree-based Representation of LCL

The representation of class probability in the diagram of LCLT is as follows:

$$\log(\hat{p}(C|\mathbf{A})) = \log(\hat{p}(C|\mathbf{A}_p)) + \log(\hat{p}(\mathbf{A}_1|C, \mathbf{A}_p)) - \log(\hat{p}(\mathbf{A}_1|\mathbf{A}_p)). \quad (14)$$

\mathbf{A}_p divides the sample set at current node into several subsets. All decomposed terms are the conditional probabilities of \mathbf{A}_p . $\hat{p}(C|\mathbf{A}_p)$ is the conditional probability on the path attributes. $\hat{p}(\mathbf{A}_1|C, \mathbf{A}_p)$ is the Naïve Bayes model at a leaf. And $\hat{p}(\mathbf{A}_1|\mathbf{A}_p)$ is the joint probability of \mathbf{A}_1 under condition of \mathbf{A}_p .

In each splitting iteration, LCL is calculated based on Eq. (14). Assuming that A_{li} denotes a leaf attribute, $\hat{p}(C|\mathbf{A}_p)$ is calculated by the ratio of the number of training samples that have the same class label to all the training samples at a leaf. $\hat{p}(\mathbf{A}_1|C, \mathbf{A}_p)$ can be represented by $\prod_{i=1}^m \hat{p}(A_{li}|C, \mathbf{A}_p)$ (m is the number of attributes at a leaf), and each $\hat{p}(A_{li}|C, \mathbf{A}_p)$ can be calculated by the ratio of the number of training samples that have the same attribute value of A_{li} and the same class label to the number of training samples that have the same class label. Likewise, $\hat{p}(\mathbf{A}_1|\mathbf{A}_p)$ can also be represented by $\prod_{i=1}^m \hat{p}(A_{li}|\mathbf{A}_p)$, and each $\hat{p}(A_{li}|\mathbf{A}_p)$ can be calculated by the ratio of the number of training samples that have the same attribute value of A_{li} to the number of training samples at that leaf.

5.2. LCLT Learning Algorithm

From the previous discussions, LCLT can represent any joint distribution. Therefore, class probability estimation based on LCLT is accurate. However, the structure learning of a LCLT can theoretically be as time-consuming as learning an optimal decision tree. A good approximation of a LCLT, which gives relatively accurate class probabilities, is desirable in many applications. Similar to a decision tree, building a LCLT can be a greedy and recursive process. We exhaustively build all possible trees in each iteration and keep only the attribute with the maximal CLL value for the next level expansion. In detail, on each iteration, choose the “best” attribute, which has the maximal CLL value, as the root of the (sub) tree, split the associated sample set into disjoint subsets corresponding to the values of that

attribute, and recur this process in each subset until certain criteria are satisfied. If the structure of a LCLT is determined, Naïve Bayes is a perfect model to represent the local conditional distribution at leaves. The algorithm is depicted in Alg. (1).

Algorithm 1 Learning Algorithm $LCLTree(\mathbf{T}, \mathbf{S}, \mathbf{A}_{\text{set}})$

T: LCL Tree

S: a set of training samples

A_{set}: a set of attributes

```

for each attribute  $A_i \in \mathbf{A}_{\text{set}}$  do
    Partition  $\mathbf{S}$  into  $\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_k$ , 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  $\mathbf{S}_i$ .
    Evaluate the split on the attribute  $A_i$  in terms of LCL.
    Choose the attribute  $A_t$  with the highest split LCL.
if the split LCL is not improved more than the LCL of attribute  $A_t$  then
    create a Naïve Bayes model at current node.
else
    for all values  $\mathbf{S}_a$  of  $A_t$  do
         $LCLTree(\mathbf{T}_a, \mathbf{S}_a, \mathbf{A}_{\text{set}} - A_t)$ .
    add  $\mathbf{T}_a$  as a child of  $\mathbf{T}$ 
Return  $\mathbf{T}$ 

```

We consider two criteria for halting growing trees. On one side, we can stop splitting when none of the alternative attributes significantly improve class probability estimation, in the form of LCL. Alternatively, to make a leaf Naïve Bayes model work accurately, there must be at least 30 training samples, which is a widely used minimum on sample size for statistical inference purposes, at current leaf. We define a split to be significant if the relative reduction in LCL is greater than 5%. Note that we train a leaf Naïve Bayes model by adopting an inner 5-fold cross-validation on the sub training sample set \mathbf{S} that falls into current leaf. For example, if an attribute has 3 attribute values which will result in three leaf Naïve Bayes models, the inner 5-fold cross-validation will be run within these leaves. Furthermore, we compute LCL by putting the training samples from all the leaves together, rather than computing the LCL for each leaf separately.

It is also worth noting, however, the different biases between learning a LCLT and learning a traditional decision tree. In a decision tree, the building process is directed by the purity of the training sample set measured by information gain, and the crucial point in selecting an attribute is whether the resulting split of training samples is “pure” or not. However, such a selection strategy does not necessarily help improving class probability estimation of test samples. In building a LCLT, we intend to choose the attributes that maximize the posterior class probabilities

$\hat{p}(C|\mathbf{A})$ among training samples at current leaf as much as possible. That means, even though there may exist a high impurity at its leaves, it could still be a good LCLT.

5.3. Experimental Methodology and Result Analysis

We utilized 33 UCI sample sets in our previous experiments, but, at this time, we used all the samples from *Letter* to do cross valuations. Two groups of comparisons were performed: we compared LCLT with Bayesian models including NB and TAN; and with PETs, which are C4.4, C4.4 with bagging (C4.4-B), C4.5, C4.5 with bagging (C4.5-B) and NBT. We implemented LCLT within Weka and used the implementations of other learning models in Weka. In all experiments, we used the outputs of LCLT as the baseline for comparison.

Tab. (14) and Tab. (16) show the experimental results in terms of LCL and AUC. The corresponding summaries of *t*-test results are demonstrated in Tab. (15) and Tab. (17). Multi-class AUC was calculated by M-measure¹⁴ in our experiments. Tab. (18) and Tab. (19) display the ACC comparisons and *t*-test results respectively. Now, our observations are summarized as follows.

- (1) LCLT outperforms NBT in terms of LCL and AUC significantly, and slightly better in ACC. The *t*-test results in LCL (see Tab. (15)) show that LCLT wins in 10 sample sets, ties in 23 sample sets and fails in 0 sample sets. In AUC (see Tab. (17)), LCLT wins in 5 sample sets, ties in 27 sample sets and fails only in one. In addition, LCLT surpasses NBT in the ACC performances as well. It wins in 3 sample sets and fails in 1 sample set.
- (2) LCLT is the best among the rest of learning models in AUC. Compared with C4.4, it wins in 19 sample sets, ties in 14 sample sets and never fail. Since C4.4 is the state-of-the-art decision tree model designed specifically for yielding accurate probability-based ranking, this comparison also provides evidence to support LCLT. Compared with NB, our model also wins in 9 sample sets, ties in 21 sample sets and fails in 3 sample sets.
- (3) In terms of the average classification accuracy (see Tab. (18)), LCLT achieves the highest ACC among all learning models. Compared with NB, it wins in 11 sample sets, ties in 21 sample sets and fails in 1 sample set. The average ACC of NB is 82.82%, lower than that of LCLT. Furthermore, LCLT also outperforms TAN significantly. It wins in 6 sample sets, ties in 24 sample sets and fails in 3 sample sets. The average ACC of TAN is 84.64%, which is lower than that of LCLT as well. At last, LCLT is also better in 8 sample sets than C4.5, the implementation of traditional decision trees.
- (4) Although C4.4 outperforms LCLT in LCL, LCLT is definitely better than C4.4 in overall performance. C4.4 sacrifices its tree size to improve class probability estimation, which could produce the “over-fitting” problem and will be noise sensitive. Therefore, in a practical perspective, LCLT is more suitable for many real applications.

The time complexity of LCLT is equivalent to NBT. Assume that t denotes the number of training samples and v represents the average number of values per attribute. In LCLT, the maximum of generated leaves is $O(t)$, and in that case the height of the tree is $O(\log_v t)$ and there are $O(t/v)$ inner nodes in the tree. At the root, LCLT performs 5-fold cross-validation on each attribute to select the best one to split and the time complexity is $O(tkn^2)$, where k is the number of class values and n is the number of attributes. As a result, the time complexity of building a complete LCLT is $O(t^2kn^2/v)$. In the classification stage, LCLT classifies a test sample by the time complexity of $O(kn)$.

6. Conclusions and Future Work

Class probability estimation provided by learning models is crucial in many applications. In this paper, we conduct a series of experiments on the performance of class probability estimation by a group of decision tree models and other non-tree models. In our study, we use *Log Conditional Likelihood* (LCL) to evaluate model performances. Experimental results show that C4.4 is the best model in LCL among other models. We point out (1) LCL is an indirect evaluation standard for class probability estimation and it can be used when true class distributions are unknown, (2) but LCL favors models that give high class probabilities. We also analyze the relations between LCL and another class probability-based metric - AUC. We conclude that no one can dominate the other. In addition, based on Naïve Bayes Tree (NBT), a hybrid model, we propose a new learning model *LCL Tree* (LCLT) to improve class probability estimation of NBT. The empirical results fulfill our expectation that, compared to other classic learning models, LCLT performs almost best in LCL, AUC and ACC.

In future research, we are going to conduct analysis on LCL based on artificial sample sets with known class distributions. This will enable us to theoretically analyze the properties of LCL in detail and make comprehensive comparisons between LCL and other metrics. We plan to design an artificial sample set where accurate class probabilities could be automatically generated. For example, the artificial set can be constructed from a Bayesian Network. The structure of Bayesian Network is defined in Fig. (3). Where the class node C is the parent of all the attribute nodes. A_1 is the attribute which has only one parent and each of other attributes $A_2 \dots A_n$ have two parents: C and its preceding attribute. We generate samples randomly by logical sampling¹⁵. Given a sample from the artificial set, its true class probabilities can easily be computed by searching the conditional probability table on each node in the network.

References

1. E. Bauer and R. Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Artificial Intelligence*, 36, 1999.
2. C. Blake and C. J. Merz. Uci repository of machine learning database.

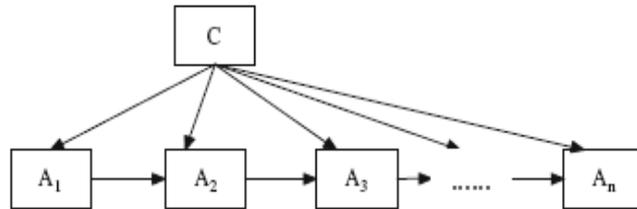


Fig. 3. Bayesian Network Structure

3. P. A. Flach C. Ferri and J. Hernandez-Orallo. Improving the auc of probabilistic estimation trees. In *Proceedings of the 14th European Conference on Machine Learning*. Springer, 2003.
4. J. Huang C. X. Ling and H. Zhang. Auc: A better measure than accuracy in comparing learning algorithms. In *Proceedings of the 16th Canadian Conference on Artificial Intelligence*. Springer, 2003.
5. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20, 1995.
6. A.P. Dawid. Calibration-based empirical probability (with discussion). *Annals of Statistics*, 13, 1985.
7. C. Elkan. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, 1991.
8. T. Fawcett F. J. Provost and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the 15th International Conference on Machine Learning*. Morgan Kaufmann, 1998.
9. N. Friedman and M. Goldszmidt. Learning bayesian networks with local structure. In *Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1996.
10. D. Grossman and P. Domingos. Learning bayesian network classifiers by maximizing conditional log likelihood. In *Proceedings of the 21th International Conference on Machine Learning*. Morgan Kaufmann, 2004.
11. Y. Guo and R. Greiner. Discriminative model selection for belief net structures. In *Proceedings of the 20th National Conference on Artificial Intelligence*. AAAI Press, 2005.
12. L. Jiang H. Zhang and J. Su. Augmenting naive bayes for ranking. In *Proceedings of the 22th International Conference on Machine Learning*. Morgan Kaufmann, 2005.
13. L. Jiang H. Zhang and J. Su. Hidden naive bayes. In *Proceedings of the 20th National Conference on Artificial Intelligence*. AAAI Press, 2005.
14. D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45, 2001.
15. M. Henrion. Propagating uncertainty in bayesian networks by probabilistic logic sampling. In *Uncertainty in Artificial Intelligence*, 1988.
16. J. Huang and C.X. Ling. Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3), 2005.
17. R. Kohavi. Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, 1996.
18. L. Kurgan and K. L. Cios. Caim discretization algorithm. *IEEE Transactions on*

- Knowledge and Data Engineering*, 16, 2004.
19. H. Liang, H. Zhang, and Y. Yan. Decision trees for probability estimation: An empirical study. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence*, 2006.
 20. C. X. Ling and R. J. Yan. Decision tree with better ranking. In *Proceedings of the 20th International Conference on Machine Learning*. Morgan Kaufmann, 2003.
 21. P. McCullagh and J. Nelder. *Generalized Linear Models*. Chapman and Hall, 1989.
 22. D. Geiger N. Friedman and M. Goldszmidt. Bayesian network classifiers. *Machine Learning*, 29, 1997.
 23. C. Nadeau and Y. Bengio. Inference for the generalization error. *Machine Learning*, 52(40), 2003.
 24. J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.
 25. F. J. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(30), 2003.
 26. J. R. Quinlan. Induction of decision trees. *Machine Learning*, 2(1), 1986.
 27. J. R. Quinlan. C4.5: Programs for machine learning. 1993.
 28. K. Spackman. Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the 6th International Workshop on Machine Learning*, 1989.
 29. J. Su and H. Zhang. Representing conditional independence using decision trees. In *Proceedings of the 20th National Conference on Artificial Intelligence*. AAAI Press, 2005.
 30. C. J. Lin T. F. Wu and R. C. Weng. Probability estimates for multi-class classification by pairwise coupling. *Machine Learning*, 5, 2004.
 31. I. H. Witten and E. Frank. *Data Mining – Practical Machine Learning Tools and Techniques with Java Implementation*. Morgan Kaufmann, 2000.
 32. H. Zhang and J. Su. Conditional independence trees. In *Proceedings of the 15th European Conference on Machine Learning*. Springer, 2004.

Table 2. Experimental results of C4.4 versus C4.5 with *Laplace* correction (C4.5-L), C4.5 with *m*-Branch (C4.5-M), C4.5 with the confusion factor algorithm (CFT4.5) and bagging with C4.5-L (C4.5-L&B): Log Conditional Likelihood (LCL) & standard deviation.

Sample Set	C4.4	C4.5-L	C4.5-M	CFT4.5	C4.5-L&B
anneal	-7.84±2.58	-8.49±3.22	-8.99±4.98	-73.52±1.70 ●	-12.24±1.85 ●
anneal.ORIG	-22.17±3.74	-25.19±3.95 ●	-25.24±5.74 ●	-80.40±1.79 ●	-34.24±4.07 ●
audiology	-15.37±3.39	-17.23±3.63 ●	-24.38±9.27 ●	-62.93±1.74 ●	-32.89±3.24 ●
autos	-13.14±2.31	-14.36±2.58 ●	-15.84±2.87 ●	-33.61±1.07 ●	-23.60±2.27 ●
balance-scale	-52.78±4.03	-56.05±3.74 ●	-56.36±3.47 ●	-56.00±0.73 ●	-45.34±2.74 ○
breast-cancer	-18.56±2.70	-16.27±1.84 ○	-16.20±1.74 ○	-17.07±0.81	-16.31±1.82 ○
breast-w	-11.17±3.39	-12.10±4.64	-13.06±4.34	-49.01±2.00 ●	-9.78±3.12
colic	-17.80±4.31	-15.53±3.98 ○	-15.32±3.85 ○	-21.61±0.97	-14.88±3.75 ○
colic.ORIG	-17.66±3.19	-16.25±2.83	-16.06±2.38 ○	-22.79±0.60 ●	-15.26±2.39 ○
credit-a	-28.06±4.92	-25.88±5.08	-25.46±5.01 ○	-41.18±2.96 ●	-24.26±5.01 ○
credit-g	-61.03±5.85	-56.37±4.20 ○	-55.20±3.66 ○	-66.08±1.52	-52.63±4.04 ○
diabetes	-43.05±4.79	-41.09±5.25	-40.67±4.71	-50.00±0.65 ●	-39.20±4.83 ○
glass	-21.02±2.70	-21.71±2.64	-22.77±3.18	-34.41±1.05 ●	-27.26±1.99 ●
heart-c	-15.85±3.68	-15.16±3.21	-15.28±3.15	-28.29±1.14	-18.68±2.85 ●
heart-h	-14.78±3.16	-14.66±3.27	-14.57±3.32	-24.52±3.89 ●	-16.03±3.20
heart-statlog	-14.00±3.33	-13.09±3.49	-12.94±3.37	-17.24±0.31 ●	-12.37±2.50
hepatitis	-6.81±2.51	-7.22±2.02	-7.08±1.81	-7.98±0.76	-6.39±1.81
hypothyroid	-90.14±5.73	-107.41±6.68 ●	-108.42±6.69 ●	-410.47±17.81●	-98.88±7.46 ●
ionosphere	-10.77±3.04	-11.42±3.04	-11.96±2.77	-22.62±0.39 ●	-9.58±2.42
iris	-3.63±1.35	-3.59±1.38	-3.97±1.29 ●	-14.10±0.22 ●	-3.70±1.26
kr-vs-kp	-8.65±3.50	-10.00±3.93	-11.21±3.95 ●	-180.50±1.77 ●	-9.01±3.15
labor	-2.22±1.28	-2.20±1.29	-2.43±1.17	-3.39±0.44 ●	-2.26±1.20
letter-2000	-193.65±10.88	-221.99±10.86●	-299.86±18.93●	-630.29±0.90 ●	-434.10±10.82●
lymph	-7.75±2.64	-7.69±2.80	-8.13±2.90	-14.14±0.90 ●	-8.79±2.12
mushroom	-2.10±0.19	-2.10±0.19	-3.13±0.45 ●	-432.76±1.84 ●	-2.18±0.20 ●
primary-tumor	-50.98±3.70	-55.94±4.41 ●	-76.43±10.23●	-93.27±1.39 ●	-79.81±3.67 ●
segment	-48.76±7.07	-55.68±7.96 ●	-58.87±9.17 ●	-401.69±3.86 ●	-85.44±6.63 ●
sick	-21.10±5.56	-26.75±8.37 ●	-26.40±8.41 ●	-154.22±9.66 ●	-25.91±8.29 ●
sonar	-11.91±2.45	-12.54±2.48	-12.20±2.15	-14.05±0.37	-10.92±1.72
soybean	-18.39±3.31	-19.28±3.53	-21.01±4.12 ●	-159.54±2.24 ●	-56.99±4.32 ●
splice	-66.48±8.24	-66.02±10.77	-70.25±11.45	-245.95±2.01 ●	-68.80±8.25
vehicle	-55.24±4.50	-57.28±4.96	-61.25±5.29 ●	-108.24±1.10 ●	-64.57±3.42 ●
vote	-6.90±3.56	-6.09±3.41 ○	-6.11±3.37	-18.16±1.00 ●	-6.09±3.22 ○
vowel	-71.55±6.18	-81.10±6.40 ●	-96.68±8.20 ●	-226.93±0.62 ●	-144.03±5.32 ●
waveform-5000	-318.55±12.98	-306.45±14.10○	-308.21±14.19○	-512.36±1.70 ●	-307.92±12.02○
zoo	-2.74±1.28	-2.90±1.30	-3.35±1.68	-13.35±0.68 ●	-4.55±1.02 ●
average	-38.13±4.11	-39.81±4.37	-43.76±5.09	-120.63±2.02	-50.69±3.83

●, ○ statistically significant degradation or improvement compared with C4.4

Table 3. *t*-test of LCL experimental results on C4.4, C4.5-L, C4.5-M, CFT4.5 and C4.5-L&B.

Models	C4.5-L	C4.5-M	CFT4.5	C4.5-L&B
C4.5-M	2/18/16			
CFT4.5	0/4/32	0/1/35		
C4.5-L&B	8/12/16	9/14/13	35/1/0	
C4.4	10/21/5	15/15/6	34/2/0	16/11/9

Table 4. Experimental results of C4.4 versus C4.5 with *Laplace* correction (C4.5-L), C4.5 with *m*-Branch (C4.5-M), C4.5 with the confusion factor algorithm (CFT4.5) and bagging with C4.5-L (C4.5-L&B): Area Under the ROC Curve (AUC) & standard deviation.

Sample Set	C4.4	C4.5-L	C4.5-M	CFT4.5	C4.5-L&B
anneal	93.67±6.25	89.12±4.34 ●	89.48±5.93 ●	88.73±5.57 ●	92.72±5.41
anneal.ORIG	91.01±8.07	89.23±6.61	90.07±7.77	89.46±5.08	90.30±7.86
audiology	64.04±2.38	62.04±2.36 ●	62.70±2.42	69.37±1.83 ○	67.05±2.34 ○
autos	91.33±4.13	91.98±3.78	93.59±2.65 ○	89.78±2.46	94.71±2.43 ○
balance-scale	61.40±6.73	56.56±8.43	55.04±7.03 ●	65.87±7.50	66.34±6.24 ○
breast-cancer	60.53±10.08	62.26±9.25	62.34±9.22	67.40±10.88	65.92±11.61
breast-w	98.22±1.25	97.44±2.24	97.40±2.26	98.30±1.88	98.77±1.17
colic	83.96±7.41	84.86±7.06	85.45±6.70	83.74±7.53	87.11±6.61 ○
colic.ORIG	83.00±6.55	84.38±6.34	85.13±6.03	81.27±7.78	87.32±5.70 ○
credit-a	89.59±3.81	89.94±3.74	90.09±3.68	90.14±3.85	91.66±3.36 ○
credit-g	70.07±4.70	72.18±4.47	72.67±4.51	73.14±5.09	75.48±5.01 ○
diabetes	76.20±5.10	77.59±6.51	77.95±6.39	77.35±6.31	80.59±5.71 ○
glass	80.11±6.91	80.00±5.96	80.36±6.07	80.10±7.09	83.40±6.01
heart-c	83.27±0.75	83.25±0.65	83.27±0.66	83.73±0.60 ○	83.70±0.61 ○
heart-h	83.30±0.64	81.15±4.59	81.18±4.60	81.72±4.02	83.82±0.63 ○
heart-statlog	82.81±8.28	84.91±7.88	85.03±7.92	86.79±6.77	87.00±6.71
hepatitis	79.50±14.28	70.16±14.77	70.31±14.85	72.08±15.64	81.78±13.28
hypothyroid	80.62±8.24	63.68±6.65 ●	68.25±6.74 ●	74.04±9.31 ●	76.55±7.92 ●
ionosphere	93.43±4.44	89.87±5.15 ●	90.14±5.01 ●	84.64±7.80 ●	94.82±3.98
iris	98.67±1.98	98.58±2.09	98.58±2.09	98.72±2.08	98.81±2.02
kr-vs-kp	99.93±0.08	99.88±0.12	99.88±0.12	98.82±0.49 ●	99.93±0.08
labor	87.17±18.05	84.15±18.11	84.15±18.11	84.02±18.81	85.83±17.79
letter-2000	85.26±2.05	85.84±1.92	92.08±1.27 ○	90.81±1.34 ○	93.50±1.24 ○
lymph	86.30±4.58	86.21±5.30	86.29±5.31	87.16±4.04	88.22±3.30
mushroom	100.00±0.00	100.00±0.00	100.00±0.00	99.46±0.18 ●	100.00±0.00
primary-tumor	68.53±3.05	68.38±3.04	70.79±2.94 ○	73.90±3.59 ○	73.31±2.89 ○
segment	99.08±0.42	98.88±0.45 ●	98.94±0.43	95.36±0.98 ●	99.34±0.32 ○
sick	99.03±0.66	93.79±4.19 ●	94.43±3.68 ●	93.98±4.23 ●	93.71±4.22 ●
sonar	78.38±9.04	73.80±10.90	74.16±10.82	72.20±10.04	81.99±9.07
soybean	98.02±1.62	97.89±1.64	98.94±1.36 ○	99.74±0.41 ○	98.97±1.29 ○
splice	98.06±0.71	98.09±0.84	98.17±0.60	98.75±0.51 ○	98.70±0.57 ○
vehicle	85.96±2.75	87.02±2.70	87.32±2.58	81.07±3.20 ●	89.32±2.14 ○
vote	97.43±2.37	97.59±2.26	97.59±2.26	98.29±1.98	97.82±2.22
vowel	91.57±2.34	90.56±2.50	95.63±1.43 ○	92.13±1.55	96.33±1.54 ○
waveform-5000	81.36±1.41	86.95±1.20 ○	88.54±1.16 ○	87.72±1.44 ○	90.52±1.22 ○
zoo	80.26±6.35	80.10±6.28	80.29±5.91	85.44±4.42 ○	80.81±6.70
average	85.59±4.65	84.40±4.84	85.17±4.74	85.42±4.90	87.67±4.42

●, ○ statistically significant degradation or improvement compared with C4.4

Table 5. *t*-test of AUC experimental results on C4.4, C4.5-L, C4.5-M, CFT4.5 and C4.5-L&B.

Models	C4.5-L	C4.5-M	CFT4.5	C4.5-L&B
C4.5-M	8/28/0			
CFT4.5	10/21/5	6/21/9		
C4.5-L&B	22/14/0	18/18/0	13/21/2	
C4.4	6/29/1	5/25/6	8/20/8	2/16/18

Table 6. Experimental results of C4.4 versus C4.4 with m -Branch (C4.4-M), C4.4 with the confusion factor algorithm (CFT4.4) and bagging with C4.4 (C4.4-B): Log Conditional Likelihood (LCL) & standard deviation.

Sample Set	C4.4	C4.4-M	CFT4.4	C4.4-B
anneal	-7.84±2.58	-8.55±4.96	-74.37±2.16●	-13.74±1.78●
anneal.ORIG	-22.17±3.74	-22.97±5.44	-87.94±1.47●	-40.13±4.44●
audiology	-15.37±3.39	-23.23±8.82 ●	-63.14±1.59●	-35.95±3.02●
autos	-13.14±2.31	-15.46±2.61 ●	-33.76±1.09●	-24.35±2.13●
balance-scale	-52.78±4.03	-53.26±3.37	-54.50±0.66	-46.71±2.46○
breast-cancer	-18.56±2.70	-17.59±2.57 ○	-17.25±0.61	-17.07±2.13○
breast-w	-11.17±3.39	-12.42±3.92	-44.06±1.81●	-10.13±2.50
colic	-17.80±4.31	-16.59±5.80	-21.39±0.69●	-15.18±3.36○
colic.ORIG	-17.66±3.19	-16.71±2.75 ○	-22.60±0.45●	-16.09±2.26○
credit-a	-28.06±4.92	-25.75±6.40 ○	-38.58±1.06●	-26.58±4.14
credit-g	-61.03±5.85	-57.22±5.97 ○	-61.95±0.62	-53.68±3.87○
diabetes	-43.05±4.79	-40.71±4.77 ○	-49.36±0.45●	-40.19±4.08○
glass	-21.02±2.70	-22.45±3.27	-34.16±0.99●	-29.77±1.98●
heart-c	-15.85±3.68	-15.46±4.16	-30.54±0.77●	-25.93±2.79●
heart-h	-14.78±3.16	-14.00±4.10	-30.15±1.03●	-24.12±3.09●
heart-statlog	-14.00±3.33	-13.09±3.76	-16.63±0.37●	-12.61±2.15○
hepatitis	-6.81±2.51	-6.87±2.78	-8.58±0.59●	-6.20±1.64
hypothyroid	-90.14±5.73	-92.18±13.03	-229.80±4.85●	-104.87±5.70●
ionosphere	-10.77±3.04	-11.17±3.31	-21.61±0.37●	-9.42±2.08○
iris	-3.63±1.35	-4.12±1.29 ●	-15.40±0.25●	-4.01±1.23●
kr-vs-kp	-8.65±3.50	-9.82±3.55 ●	-182.42±1.80●	-7.92±2.75
labor	-2.22±1.28	-2.32±1.34	-3.16±0.44●	-2.13±0.95
letter-2000	-193.65±10.88	-296.32±18.62●	-627.85±0.90●	-454.15±9.78●
lymph	-7.75±2.64	-7.91±2.85	-13.91±0.88●	-9.85±1.85●
mushroom	-2.10±0.19	-3.13±0.45 ●	-432.76±1.84●	-2.18±0.20●
primary-tumor	-50.98±3.70	-75.42±9.65 ●	-95.96±1.11●	-82.41±3.45●
segment	-48.76±7.07	-55.80±8.75 ●	-406.77±1.96●	-97.61±6.49●
sick	-21.10±5.56	-19.38±6.43 ○	-152.99±3.69●	-19.66±4.67
sonar	-11.91±2.45	-11.50±2.28	-13.88±0.39●	-10.76±1.50
soybean	-18.39±3.31	-20.72±4.13 ●	-164.86±1.71●	-61.37±4.69●
splice	-66.48±8.24	-70.78±10.83●	-250.19±1.64●	-78.71±6.91●
vehicle	-55.24±4.50	-62.53±5.14 ●	-107.69±1.08●	-70.21±3.09●
vote	-6.90±3.56	-6.67±4.10	-19.30±0.78●	-6.10±3.25
vowel	-71.55±6.18	-95.45±8.10 ●	-226.45±0.63●	-152.25±4.88●
waveform-5000	-318.55±12.98	-329.01±13.84●	-509.67±1.04●	-351.30±9.38●
zoo	-2.74±1.28	-3.18±1.65	-13.76±0.59●	-4.59±1.00●
average	-38.13±4.11	-43.33±5.41	-118.96±1.18	-54.66±3.38

●, ○ statistically significant degradation or improvement compared with C4.4

Table 7. t -test of LCL experimental results on C4.4, C4.4-M, CFT4.4 and C4.4-B.

Models	C4.4-M	CFT4.4	C4.4-B
CFT4.4	0/3/33		
C4.4-B	6/12/18	35/1/0	
C4.4	13/17/6	33/3/0	20/8/8

Table 8. Experimental results of C4.4 versus C4.4 with *m*-Branch (C4.4-M), C4.4 with the confusion factor algorithm (CFT4.4) and bagging with C4.4 (C4.4-B): Area Under the ROC Curve (AUC) & standard deviation.

Sample Set	C4.4	C4.4-M	CFT4.4	C4.4-B
anneal	93.67±6.25	93.74±6.10	94.14±3.14	94.48±5.78
anneal.ORIG	91.01±8.07	92.07±8.27 ◦	92.40±6.49	93.21±7.91 ◦
audiology	64.04±2.38	64.96±2.20 ◦	70.45±1.24 ◦	69.05±1.84 ◦
autos	91.33±4.13	94.06±2.37 ◦	90.42±2.76	94.90±2.36 ◦
balance-scale	61.40±6.73	56.35±6.06 ●	66.69±5.63 ◦	66.55±6.03 ◦
breast-cancer	60.53±10.08	61.85±10.78	67.35±11.38◦	64.55±10.49
breast-w	98.22±1.25	98.23±1.26	98.35±1.35	98.83±1.01 ◦
colic	83.96±7.41	87.00±7.30 ◦	85.90±6.60	88.20±6.63 ◦
colic.ORIG	83.00±6.55	83.98±6.16	80.82±7.60	85.98±5.38
credit-a	89.59±3.81	91.26±3.57 ◦	91.15±3.58	90.53±3.59
credit-g	70.07±4.70	73.48±4.41 ◦	75.41±4.66 ◦	74.21±4.74 ◦
diabetes	76.20±5.10	78.95±5.35 ◦	79.81±6.19 ◦	79.10±5.45 ◦
glass	80.11±6.91	81.57±6.56	82.32±5.12	80.30±7.12
heart-c	83.27±0.75	83.46±0.70	83.82±0.56 ◦	83.65±0.64 ◦
heart-h	83.30±0.64	83.66±0.63 ◦	83.78±0.64 ◦	83.64±0.62 ◦
heart-statlog	82.81±8.28	85.29±7.98	88.76±6.22 ◦	86.51±6.67 ◦
hepatitis	79.50±14.28	80.93±13.76	77.92±16.41	82.43±13.79
hypothyroid	80.62±8.24	85.64±7.81 ◦	81.52±7.31	81.44±7.56
ionosphere	93.43±4.44	93.11±4.68	91.42±5.10	95.51±3.58 ◦
iris	98.67±1.98	98.69±1.99	97.37±3.44	98.77±2.09
kr-vs-kp	99.93±0.08	99.93±0.08	98.73±0.52 ●	99.97±0.05
labor	87.17±18.05	89.35±16.49	85.98±19.88	90.79±15.51
letter-2000	85.26±2.05	92.59±1.25 ◦	91.21±1.39 ◦	93.73±1.25 ◦
lymph	86.30±4.58	87.03±4.49	87.13±4.82	88.17±3.11
mushroom	100.00±0.00	100.00±0.00	99.46±0.18 ●	100.00±0.00
primary-tumor	68.53±3.05	71.59±2.89 ◦	74.63±3.51 ◦	73.05±3.07 ◦
segment	99.08±0.42	99.19±0.39	95.87±1.08 ●	99.49±0.28 ◦
sick	99.03±0.66	99.16±0.62	94.40±4.08 ●	99.23±0.46
sonar	78.38±9.04	79.24±9.15	75.32±10.51	83.43±8.73
soybean	98.02±1.62	98.94±1.36 ◦	99.54±0.33 ◦	98.95±1.26 ◦
splice	98.06±0.71	98.49±0.64 ◦	98.94±0.43 ◦	98.74±0.58 ◦
vehicle	85.96±2.75	87.34±2.57 ◦	82.00±3.00 ●	89.02±2.16 ◦
vote	97.43±2.37	97.49±2.38	98.81±1.34 ◦	98.31±2.18
vowel	91.57±2.34	96.07±1.41 ◦	93.05±1.57	96.44±1.58 ◦
waveform-5000	81.36±1.41	87.15±1.21 ◦	90.77±1.07 ◦	90.04±1.28 ◦
zoo	80.26±6.35	80.52±6.00	86.46±4.26 ◦	80.88±6.71
average	85.59±4.65	87.01±4.41	87.00±4.54	88.11±4.21

●, ◦ statistically significant degradation or improvement compared with C4.4

Table 9. t-test of AUC experimental results on C4.4, C4.4-M, CFT4.4 and C4.4-B.

Models	C4.4-M	CFT4.4	C4.4-B
CFT4.4	9/18/9		
C4.4-B	9/26/1	11/23/2	
C4.4	1/19/16	5/16/15	0/16/20

Table 10. Experimental results of C4.4 versus Naïve Bayes Tree (NBT), Naïve Bayes (NB), Tree-Augmented Naïve Bayes (TAN), 5 Nearest Neighbors (KNN-5) and Support Vector Machine (SVM): Log Conditional Likelihood (LCL) & standard deviation.

Sample Set	C4.4	NBT	NB	TAN	KNN-5	SVM
anneal	-7.84±2.58	-18.46±17.63	-14.22±6.16 ●	-6.29±5.36	-8.22±2.80	-3.52±5.32 ○
anneal.ORIG	-22.17±3.74	-33.33±16.32●	-23.58±5.60	-19.55±6.90	-27.40±5.44 ●	-23.25±6.33
audiology	-15.37±3.39	-95.28±41.89●	-65.91±24.28●	-67.19±24.11●	-31.61±7.73 ●	-39.12±24.07●
autos	-13.14±2.31	-34.94±16.81●	-45.57±18.12●	-33.91±17.06●	-19.82±6.45 ●	-23.74±11.09●
balance-scale	-52.78±4.03	-31.75±1.51 ○	-31.75±1.51 ○	-34.78±3.10 ○	-67.11±2.71 ●	-14.04±3.99 ○
breast-cancer	-18.56±2.70	-20.47±5.23	-18.37±4.49	-18.17±3.60	-18.75±3.97	-17.47±2.83
breast-w	-11.17±3.39	-17.47±13.63	-18.28±14.16	-12.14±6.76	-9.75±5.08	-11.87±7.79
colic	-17.80±4.31	-34.42±17.34●	-30.63±11.38●	-26.22±9.35 ●	-19.04±6.30	-22.49±8.02 ●
colic.ORIG	-17.66±3.19	-38.50±17.60●	-21.24±5.74	-22.36±6.24 ●	-25.19±6.18 ●	-30.58±11.15●
credit-a	-28.06±4.92	-34.52±11.89	-28.79±8.10	-28.07±7.06	-29.82±7.91	-27.17±5.66
credit-g	-61.03±5.85	-62.44±23.22	-52.79±6.35 ○	-56.16±8.09	-63.26±9.89	-52.16±5.68 ○
diabetes	-43.05±4.79	-42.70±9.11	-40.78±7.49	-42.51±8.23	-45.44±7.22	-39.88±6.09
glass	-21.02±2.70	-31.06±9.62 ●	-24.08±5.42	-26.15±6.27 ●	-23.54±5.89	-25.14±7.38
heart-c	-15.85±3.68	-15.70±7.49	-13.91±6.71	-14.01±6.09	-13.97±5.46	-13.45±5.45
heart-h	-14.78±3.16	-14.73±5.94	-13.49±5.37	-12.96±4.06	-13.41±4.57	-13.21±4.59
heart-statlog	-14.00±3.33	-16.31±9.29	-12.25±4.96	-14.60±5.39	-11.68±3.61	-12.76±5.10
hepatitis	-6.81±2.51	-9.18±5.78	-8.53±5.98	-8.16±4.72	-7.20±3.69	-10.74±8.22
hypothyroid	-90.14±5.73	-98.23±14.58	-97.14±13.29	-93.72±12.69	-131.25±20.97●	-94.62±13.50
ionosphere	-10.77±3.04	-35.54±20.03●	-34.79±19.94●	-18.17±13.24	-13.45±7.42	-171.50±96.25●
iris	-3.63±1.35	-2.69±2.90	-2.56±2.35	-3.12±2.30	-3.04±2.25	-2.59±2.88
kr-vs-kp	-8.65±3.50	-28.01±18.07●	-93.48±7.65 ●	-60.27±7.38 ●	-58.41±6.49 ●	-37.71±8.08 ●
labor	-2.22±1.28	-1.03±2.27	-0.71±0.99 ○	-2.23±3.43	-1.61±0.90	-6.43±9.89
letter-2000	-193.65±10.88	-382.03±50.64●	-299.04±29.19●	-260.71±33.34●	-297.17±32.04●	-221.59±27.54●
lymph	-7.75±2.64	-8.48±5.51	-6.22±3.96	-7.15±5.24	-6.90±3.21	-9.10±5.67
mushroom	-2.10±0.19	-0.14±0.14 ○	-105.77±23.25●	-0.19±0.45 ○	-0.05±0.34 ○	-0.00±0.00 ○
primary-tumor	-50.98±3.70	-74.19±14.56●	-65.56±8.27 ●	-69.75±8.85 ●	-93.51±12.35●	-81.10±16.98●
segment	-48.76±7.07	-111.94±45.14●	-124.32±33.74●	-40.15±13.46○	-58.30±12.72●	-37.99±12.83○
sick	-21.10±5.56	-45.55±19.82●	-46.05±11.99●	-28.91±8.80 ●	-27.64±8.62 ●	-33.45±10.45●
sonar	-11.91±2.45	-38.85±19.05●	-22.67±11.47●	-28.73±13.48●	-8.90±2.89 ○	-178.12±92.92●
soybean	-18.39±3.31	-28.63±15.19●	-26.25±11.03●	-8.06±3.84 ○	-16.67±5.16	-15.44±6.18
splice	-66.48±8.24	-47.11±13.57○	-46.53±12.85○	-46.89±11.95○	-181.79±19.56●	-126.34±54.97●
vehicle	-55.24±4.50	-137.97±32.69●	-172.12±27.55●	-57.52±10.16	-61.21±9.89	-68.11±11.51●
vote	-6.90±3.56	-7.35±5.41	-27.25±13.85●	-7.91±5.39	-10.94±7.44	-5.55±3.89
vowel	-71.55±6.18	-45.93±16.44○	-89.80±11.38●	-21.87±8.84 ○	-62.71±7.64 ○	-55.20±16.88○
waveform-5000	-318.55±12.98	-309.13±43.99	-378.00±32.64●	-254.80±23.42○	-305.25±18.78	-232.69±24.93○
zoo	-2.74±1.28	-1.29±1.68 ○	-1.22±1.06 ○	-1.07±1.44 ○	-1.64±0.95 ○	-2.29±3.41
average	-38.13±4.11	-54.32±15.89	-58.43±11.62	-40.40±8.89	-49.32±7.63	-48.90±15.21

●, ○ statistically significant degradation or improvement compared with C4.4

Table 11. *t*-test of LCL experimental results on C4.4, NBT, NB, TAN, KNN-5 and SVM.

Models	SVM	KNN-5	TAN	NB	NBT
KNN-5	3/20/13				
TAN	8/21/7	10/22/4			
NB	5/16/15	7/13/16	3/19/16		
NBT	3/19/14	6/18/12	2/22/12	6/24/6	
C4.4	12/17/7	12/20/4	10/19/7	17/14/5	15/16/5

Table 12. Experimental results of C4.4 versus Naïve Bayes Tree (NBT), Naïve Bayes (NB), Tree-Augmented Naïve Bayes (TAN), 5 Nearest Neighbors (KNN-5) and Support Vector Machine (SVM): Area Under the ROC Curve (AUC) & standard deviation.

Sample Set	C4.4	NBT	NB	TAN	KNN-5	SVM
anneal	93.67±6.25	96.31±1.10	96.18±1.07	96.59±0.13	94.98±5.01	96.31±1.06
anneal.ORIG	91.01±8.07	93.55±7.60 ◦	94.50±4.03 ◦	95.26±2.80 ◦	94.26±5.71	95.33±2.05
audiology	64.04±2.38	70.17±1.26 ◦	70.02±1.09 ◦	70.25±1.05 ◦	69.32±1.33 ◦	70.79±0.94 ◦
autos	91.33±4.13	94.11±2.72 ◦	91.54±3.61	94.37±2.46 ◦	90.24±4.58	95.01±2.26 ◦
balance-scale	61.40±6.73	84.64±4.34 ◦	84.64±4.34 ◦	78.34±5.04 ◦	66.12±3.36 ◦	95.72±2.63 ◦
breast-cancer	60.53±10.08	67.98±10.28◦	70.18±10.88◦	66.18±10.71	65.96±10.13	64.98±12.20
breast-w	98.22±1.25	99.25±0.72 ◦	99.25±0.73 ◦	98.72±1.04	98.73±1.33	98.73±1.13
colic	83.96±7.41	86.78±7.24	84.36±6.95	85.04±6.45	85.85±6.10	84.68±6.62
colic.ORIG	83.00±6.55	79.83±8.33	81.18±7.47	81.93±6.93	73.66±8.78 ●	80.69±6.99
credit-a	89.59±3.81	91.34±3.25 ◦	91.86±3.11 ◦	91.35±3.21	91.63±3.33 ◦	90.57±3.45
credit-g	70.07±4.70	77.53±5.24 ◦	79.10±4.14 ◦	77.92±4.82 ◦	73.64±4.12	78.03±4.47 ◦
diabetes	76.20±5.10	82.11±5.06 ◦	82.61±4.97 ◦	81.33±5.19 ◦	76.79±5.17	81.64±5.23 ◦
glass	80.11±6.91	79.13±6.39	78.42±6.03	78.32±6.12	81.85±5.35	86.24±5.07 ◦
heart-c	83.27±0.75	84.00±0.58 ◦	84.11±0.56 ◦	84.03±0.59 ◦	83.97±0.58 ◦	83.97±0.60 ◦
heart-h	83.30±0.64	83.90±0.59 ◦	84.00±0.56 ◦	83.88±0.55 ◦	83.78±0.62 ◦	83.77±0.66 ◦
heart-statlog	82.81±8.28	89.83±5.82 ◦	91.34±4.85 ◦	88.19±5.75	89.98±4.90 ◦	89.64±5.66 ◦
hepatitis	79.50±14.28	85.69±11.66	89.36±10.03◦	86.06±10.46	83.40±12.20	83.37±12.25
hypothyroid	80.62±8.24	87.66±6.75 ◦	88.10±6.49 ◦	87.84±7.10 ◦	84.02±7.87 ◦	86.60±7.36 ◦
ionosphere	93.43±4.44	94.31±4.22	93.69±4.57	98.08±2.17 ◦	92.98±5.32	94.48±3.66
iris	98.67±1.98	98.85±2.00	98.99±1.69	98.49±2.44	98.61±2.18	98.13±2.90
kr-vs-kp	99.93±0.08	99.44±0.60 ●	95.19±1.19 ●	98.06±0.63 ●	99.32±0.39 ●	99.13±0.40 ●
labor	87.17±18.05	96.63±11.83	98.67±5.39	93.75±14.34	96.44±7.32	94.21±13.54
letter-2000	85.26±2.05	94.78±1.05 ◦	94.95±0.83 ◦	96.40±0.81 ◦	94.01±1.30 ◦	96.87±0.79 ◦
lymph	86.30±4.58	88.94±2.81	90.25±1.57 ◦	89.16±3.28	88.63±3.25	88.47±3.94
mushroom	100.00±0.00	100.00±0.00	99.80±0.07 ●	100.00±0.00	100.00±0.00	100.00±0.00
primary-tumor	68.53±3.05	74.71±2.75 ◦	75.58±2.75 ◦	75.43±2.67 ◦	72.21±3.23 ◦	75.34±2.74 ◦
segment	99.08±0.42	99.11±0.33	98.35±0.45 ●	99.63±0.18 ◦	99.07±0.37	99.46±0.32 ◦
sick	99.03±0.66	94.46±3.47 ●	95.87±2.31 ●	98.31±1.04 ●	98.29±1.40	93.94±4.25 ●
sonar	78.38±9.04	79.18±9.38	85.50±8.64	82.04±9.73	89.17±7.24 ◦	82.91±10.11
soybean	98.02±1.62	99.72±0.32 ◦	99.79±0.24 ◦	99.87±0.23 ◦	99.59±0.50 ◦	99.79±0.23 ◦
splice	98.06±0.71	99.44±0.31 ◦	99.46±0.27 ◦	99.40±0.35 ◦	96.96±0.68 ●	98.01±0.79
vehicle	85.96±2.75	85.86±3.30	80.58±3.04 ●	91.14±1.89 ◦	87.85±2.44	86.66±2.81
vote	97.43±2.37	98.61±1.57	97.15±2.00	98.78±1.22 ◦	97.23±2.54	98.94±1.62 ◦
vowel	91.57±2.34	98.59±0.80 ◦	95.98±1.07 ◦	99.64±0.25 ◦	98.37±0.73 ◦	98.60±0.68 ◦
waveform-5000	81.36±1.41	93.71±0.96 ◦	95.32±0.68 ◦	93.87±0.80 ◦	89.17±1.07 ◦	93.76±0.54 ◦
zoo	80.26±6.35	89.02±2.99 ◦	88.88±2.83 ◦	88.93±2.82 ◦	88.40±3.28 ◦	88.05±4.36 ◦
average	85.59±4.65	89.42±3.82	89.58±3.35	89.63±3.48	88.18±3.71	89.80±3.73

●, ◦ statistically significant degradation or improvement compared with C4.4

Table 13. *t*-test of AUC experimental results on C4.4, NBT, NB, TAN, KNN-5 and SVM.

Models	SVM	KNN-5	TAN	NB	NBT
KNN-5	2/21/13				
TAN	7/23/6	11/21/4			
NB	4/21/11	10/19/7	4/21/11		
NBT	1/29/6	9/25/2	6/27/3	7/27/2	
C4.4	2/16/18	3/19/14	2/14/20	5/10/21	2/14/20

Table 14. Experimental results of LCLT versus Naïve Bayes Tree (NBT), Naïve Bayes (NB) and Tree-Augmented Naïve Bayes (TAN); C4.4 and C4.4 with bagging (C4.4-B): Log Conditional Likelihood (LCL) & standard deviation.

Data Set	LCLT	NBT	NB	TAN	C4.4	C4.4-B
anneal	-10.78±15.51	-18.46±17.63	-14.22±6.16	-6.29±5.36	-7.84±2.58	-13.74±1.78
anneal.ORIG	-22.28±11.36	-33.33±16.32●	-23.58±5.60	-19.55±6.90	-22.17±3.74	-40.13±4.44●
audiology	-75.58±47.43	-95.28±41.89	-65.91±24.28	-67.19±24.11	-15.37±3.39 ○	-35.95±3.02○
balance-scale	-29.81±1.51	-31.75±1.51 ●	-31.75±1.51 ●	-34.78±3.10 ●	-52.78±4.03 ●	-46.71±2.46●
breast-cancer	-18.88±4.25	-20.47±5.23	-18.37±4.49	-18.17±3.60	-18.56±2.70	-17.07±2.13
breast-w	-11.43±6.38	-17.47±13.63	-18.28±14.16	-12.14±6.76	-11.17±3.39	-10.13±2.50
colic	-30.82±13.42	-34.42±17.34	-30.63±11.38	-26.22±9.35	-17.80±4.31 ○	-15.18±3.36○
colic.ORIG	-24.96±10.60	-38.50±17.60●	-21.24±5.74	-22.36±6.24	-17.66±3.19 ○	-16.09±2.26○
credit-a	-26.98±6.93	-34.52±11.89●	-28.79±8.10	-28.07±7.06	-28.06±4.92	-26.58±4.14
credit-g	-52.61±6.18	-62.44±23.22	-52.79±6.35	-56.16±8.09	-61.03±5.85 ●	-53.68±3.87
diabetes	-40.30±7.20	-42.70±9.11	-40.78±7.49 ●	-42.51±8.23	-43.05±4.79	-40.19±4.08
glass	-26.06±8.73	-31.06±9.62	-24.08±5.42	-26.15±6.27	-21.02±2.70	-29.77±1.98
heart-c	-17.92±8.48	-15.70±7.49	-13.91±6.71	-14.01±6.09	-15.85±3.68	-25.93±2.79●
heart-h	-15.93±7.24	-14.73±5.94	-13.49±5.37	-12.96±4.06	-14.78±3.16	-24.12±3.09●
heart-statlog	-12.01±4.74	-16.31±9.29	-12.25±4.96	-14.60±5.39 ●	-14.00±3.33	-12.61±2.15
hepatitis	-9.38±5.98	-9.18±5.78	-8.53±5.98	-8.16±4.72	-6.81±2.51	-6.20±1.64
hypothyroid	-95.50±13.68	-98.23±14.58	-97.14±13.29	-93.72±12.69	-90.14±5.73	-104.87±5.70●
iris	-2.73±2.55	-2.69±2.90	-2.56±2.35	-3.12±2.30	-3.63±1.35	-4.01±1.23●
kr-vs-kp	-18.39±14.46	-28.01±18.07	-93.48±7.65 ●	-60.27±7.38 ●	-8.65±3.50 ○	-7.92±2.75○
labor	-1.50±2.90	-1.03±2.27	-0.71±0.99	-2.23±3.43	-2.22±1.28	-2.13±0.95
letter	-1853±168	-2193±159 ●	-2505±98 ●	-1272 ±68 ○	-1048±30 ○	-2927±40 ●
lymph	-9.16±7.43	-8.48±5.51	-6.22±3.96	-7.15±5.24	-7.75±2.64	-9.85±1.85
mushroom	0.00±0.01	-0.14±0.14 ●	-105.77±23.25●	-0.19±0.45	-2.10±0.19 ●	-2.18±0.20●
primary-tumor	-74.57±12.93	-74.19±14.56	-65.56±8.27 ○	-69.75±8.85	-50.98±3.70 ○	-82.41±3.45
segment	-61.82±22.64	-111.94±45.14●	-124.32±33.74●	-40.15±13.46○	-48.76±7.07	-97.61±6.49●
sick	-24.51±9.35	-45.55±19.82●	-46.05±11.99●	-28.91±8.80 ●	-21.10±5.56 ○	-19.66±4.67○
soybean	-17.39±11.73	-28.63±15.19	-26.25±11.03	-8.06±3.84 ○	-18.39±3.31	-61.37±4.69●
splice	-46.58±12.76	-47.11±13.57	-46.53±12.85	-46.89±11.95	-66.48±8.24 ●	-78.71±6.91●
vehicle	-98.66±21.48	-137.97±32.69●	-172.12±27.55●	-57.52±10.16○	-55.24±4.50 ○	-70.21±3.09○
vote	-7.78±5.33	-7.35±5.41	-27.25±13.85●	-7.91±5.39	-6.90±3.56	-6.10±3.25
vowel	-38.23±21.06	-45.93±16.44	-89.80±11.38●	-21.87±8.84 ○	-71.55±6.18 ●	-152.25±4.88●
waveform-5000	-228.39±19.84	-309.13±43.99●	-378.00±32.64●	-254.80±23.42●	-318.55±12.98●	-351.30±9.38●
zoo	-2.14±2.63	-1.29±1.68	-1.22±1.06	-1.07±1.44	-2.74±1.28	-4.59±1.00●
average	-91.09±15.60	-110.82±18.92	-127.47±13.26	-72.27±9.42	-66.40±4.83	-133.22±4.43

●, ○ statistically significant degradation or improvement compared with LCLT

Table 15. *t*-test of LCL experimental results on LCLT, NBT, NB, TAN, C4.4 and C4.4-B.

Models	C4.4-B	C4.4	TAN	NB	NBT
C4.4	19/7/7				
TAN	16/12/5	8/17/8			
NB	14/9/10	5/14/14	3/18/12		
NBT	10/15/8	5/16/12	2/20/11	7/22/4	
LCLT	14/13/6	6/19/8	5/23/5	11/21/1	10/23/0

Table 16. Experimental results of LCLT versus Naïve Bayes Tree (NBT), Naïve Bayes (NB) and Tree-Augmented Naïve Bayes (TAN); C4.4 and C4.4 with bagging (C4.4-B): Area Under the Curve (AUC) & standard deviation.

Data Set	LCLT	NBT	NB	TAN	C4.4	C4.4-B
anneal	95.97±1.33	96.31±1.10	96.18±1.07	96.59±0.13	93.67±6.25	94.48±5.78
anneal.ORIG	93.73±7.40	93.55±7.60	94.50±4.03	95.26±2.80	91.01±8.07 ●	93.21±7.91
audiology	70.36±1.17	70.17±1.26	70.02±1.09	70.25±1.05	64.04±2.38 ●	69.05±1.84 ●
balance-scale	84.69±4.31	84.64±4.34	84.64±4.34	78.34±5.04 ●	61.40±6.73 ●	66.55±6.03 ●
breast-cancer	68.00±10.45	67.98±10.28	70.18±10.88	66.18±10.71	60.53±10.08●	64.55±10.49
breast-w	98.64±1.04	99.25±0.72 ○	99.25±0.73 ○	98.72±1.04	98.22±1.25	98.83±1.01
colic	82.08±8.30	86.78±7.24	84.36±6.95	85.04±6.45	83.96±7.41	88.20±6.63 ○
colic.ORIG	81.95±7.24	79.83±8.33	81.18±7.47	81.93±6.93	83.00±6.55	85.98±5.38
credit-a	92.06±3.11	91.34±3.25	91.86±3.11	91.35±3.21	89.59±3.81 ●	90.53±3.59 ●
credit-g	79.14±4.11	77.53±5.24	79.10±4.14	77.92±4.82	70.07±4.70 ●	74.21±4.74 ●
diabetes	82.57±4.97	82.11±5.06	82.61±4.97	81.33±5.19	76.20±5.10 ●	79.10±5.45
glass	82.17±5.93	79.13±6.39	78.42±6.03 ●	78.32±6.12 ●	80.11±6.91	80.30±7.12
heart-c	83.89±0.62	84.00±0.58	84.11±0.56	84.03±0.59	83.27±0.75 ●	83.65±0.64
heart-h	83.87±0.58	83.90±0.59	84.00±0.56	83.88±0.55	83.30±0.64 ●	83.64±0.62
heart-statlog	91.34±4.84	89.83±5.82	91.34±4.85	88.19±5.75 ●	82.81±8.28 ●	86.51±6.67 ●
hepatitis	83.48±12.95	85.69±11.66	89.36±10.03	86.06±10.46	79.50±14.28	82.43±13.79
hypothyroid	88.23±6.67	87.66±6.75	88.10±6.49	87.84±7.10	80.62±8.24 ●	81.44±7.56 ●
iris	98.72±2.02	98.85±2.00	98.99±1.69	98.49±2.44	98.67±1.98	98.77±2.09
kr-vs-kp	99.82±0.20	99.44±0.60	95.19±1.19 ●	98.06±0.63 ●	99.93±0.08	99.97±0.05 ○
labor	95.29±14.62	96.63±11.83	98.67±5.39	93.75±14.34	87.17±18.05	90.79±15.51
letter	99.36±0.10	98.51±0.17 ●	96.91±0.21 ●	99.12±0.09 ●	95.52±0.32 ●	98.41±0.21 ●
lymph	89.12±2.74	88.94±2.81	90.25±1.57	89.16±3.28	86.30±4.58	88.17±3.11
mushroom	100.00±0.00	100.00±0.00	99.80±0.07 ●	100.00±0.00	100.00±0.00	100.00±0.00
primary-tumor	75.33±2.88	74.71±2.75	75.58±2.75	75.43±2.67	68.53±3.05 ●	73.05±3.07 ●
segment	99.40±0.24	99.11±0.33 ●	98.35±0.45 ●	99.63±0.18 ○	99.08±0.42 ●	99.49±0.28
sick	98.44±1.47	94.46±3.47 ●	95.87±2.31 ●	98.31±1.04	99.03±0.66	99.23±0.46
soybean	99.81±0.29	99.72±0.32	99.79±0.24	99.87±0.23	98.02±1.62 ●	98.95±1.26 ●
splice	99.45±0.27	99.44±0.31	99.46±0.27 ○	99.40±0.35	98.06±0.71 ●	98.74±0.58 ●
vehicle	86.68±2.53	85.86±3.30	80.58±3.04 ●	91.14±1.89 ○	85.96±2.75	89.02±2.16 ○
vote	98.50±1.44	98.61±1.57	97.15±2.00 ●	98.78±1.22	97.43±2.37	98.31±2.18
vowel	99.35±0.60	98.59±0.80 ●	95.98±1.07 ●	99.64±0.25	91.57±2.34 ●	96.44±1.58 ●
waveform-5000	94.74±0.67	93.71±0.96 ●	95.32±0.68 ○	93.87±0.80 ●	81.36±1.41	90.04±1.28 ●
zoo	88.64±2.95	89.02±2.99	88.88±2.83	88.93±2.82	80.26±6.35 ●	80.88±6.71 ●
average	89.83±3.58	89.55±3.65	89.58±3.12	89.54±3.34	85.70±4.49	87.97±4.11

●, ○ statistically significant degradation or improvement compared with LCLT

Table 17. *t*-test of AUC experimental results on LCLT, NBT, NB, TAN, C4.4 and C4.4-B.

Models	C4.4-B	C4.4	TAN	NB	NBT
C4.4	0/15/18				
TAN	12/19/2	18/13/2			
NB	14/12/7	21/7/5	4/20/9		
NBT	8/20/5	19/12/2	3/25/5	7/25/1	
LCLT	14/16/3	19/14/0	6/25/2	9/21/3	5/27/1

Table 18. Experimental results of LCLT versus Naïve Bayes Tree (NBT), Naïve Bayes (NB) and Tree-Augmented Naïve Bayes (TAN); C4.5, C4.5 with Laplace estimation (C4.5-L), and C4.5 with bagging (C4.5-B): Classification Accuracy (ACC) & standard deviation.

Data Set	LCLT	NBT	NB	TAN	C4.5	C4.5-B
anneal	99.06±1.00	98.40±1.53	94.32±2.23 ●	98.34±1.18	98.65±0.97	98.76±0.89
anneal.ORIG	89.94±3.33	91.27±3.03	88.16±3.06	90.88±2.55	90.36±2.51	91.78±2.44
audiology	78.40±8.63	76.66±7.47	71.40±6.37 ●	72.68±7.02 ●	77.22±7.69	80.67±7.31
balance-scale	91.44±1.30	91.44±1.30	91.44±1.30	86.22±2.82 ●	64.14±4.16 ●	73.30±5.38 ●
breast-cancer	72.14±7.19	71.66±7.92	72.94±7.71	70.09±7.68	75.26±5.04	73.09±5.75
breast-w	95.08±2.48	97.23±1.76○	97.30±1.75 ○	94.91±2.37	94.01±3.28	95.34±2.71
colic	78.08±7.38	82.50±6.51	78.86±6.05	80.57±5.90	84.31±6.02 ○	84.56±6.21 ○
colic.ORIG	75.57±6.49	74.83±7.82	74.21±7.09	76.11±6.04	80.79±5.66 ○	82.64±5.44 ○
credit-a	85.13±4.10	84.86±3.92	84.74±3.83	84.43±4.51	85.06±4.12	85.83±4.20
credit-g	76.01±3.76	75.54±3.92	75.93±3.87	75.86±3.58	72.61±3.49 ●	73.89±3.78
diabetes	75.63±4.81	75.28±4.84	75.68±4.85	75.09±4.96	73.89±4.70	73.91±4.63
glass	58.69±10.15	58.00±9.42	57.69±10.07	58.43±8.86	58.14±8.48	57.98±8.99
heart-c	80.54±6.83	81.10±7.24	83.44±6.27	82.85±7.20	79.14±6.44	79.48±6.20
heart-h	81.41±6.56	82.46±6.26	83.64±5.85	82.14±6.20	80.10±7.11	80.90±7.08
heart-statlog	83.59±5.32	82.26±6.50	83.78±5.41	79.37±6.87 ●	79.78±7.71	79.44±6.52
hepatitis	81.20±9.78	82.90±9.79	84.06±9.91	82.40±8.68	81.12±8.42	81.38±7.74
hypothyroid	92.90±0.73	93.05±0.65	92.79±0.73	93.23±0.68	93.24±0.44	93.25±0.45
iris	93.73±6.82	95.27±6.16	94.33±6.79	91.67±7.18	96.00±4.64	95.53±5.02
kr-vs-kp	98.93±0.65	97.81±2.05	87.79±1.91 ●	92.05±1.49 ●	99.44±0.37 ●	99.42±0.38 ○
labor	93.93±10.94	95.60±8.39	96.70±7.27	90.33±10.96	84.97±14.24	85.23±13.11
letter	86.24±0.72	83.49±0.81●	70.09±0.93 ●	83.11±0.75 ●	81.31±0.78 ●	83.69±0.85 ●
lymph	82.79±9.81	82.21±8.95	85.97±8.88	84.07±8.93	78.21±9.74	78.97±9.06
mushroom	100.00±0.00	100.00±0.00	95.52±0.78 ●	99.99±0.03	100.00±0.00	100.00±0.00
primary-tumor	46.17±5.90	45.84±6.61	47.20±6.02	46.76±5.92	41.01±6.59 ●	43.42±6.08
segment	93.13±1.34	92.64±1.61	89.03±1.66 ●	94.54±1.60 ○	93.42±1.67	93.97±1.46
sick	97.80±0.82	97.86±0.69	96.78±0.91 ●	97.61±0.73	98.16±0.68	98.17±0.71
soybean	93.07±2.57	92.30±2.70	92.20±3.23	95.24±2.28 ○	92.63±2.72	93.66±2.70
splice	95.39±1.15	95.42±1.14	95.42±1.14	95.39±1.16	94.17±1.28 ●	94.51±1.28
vehicle	68.83±4.01	68.91±4.58	61.03±3.48 ●	73.71±3.48 ○	70.74±3.62	71.93±4.07
vote	94.65±3.11	94.78±3.32	90.21±3.95 ●	94.57±3.23	96.27±2.79	96.32±2.65
vowel	91.59±3.53	88.01±3.71●	66.09±4.78 ●	93.10±2.85	75.57±4.58 ●	79.44±3.73 ●
waveform-5000	84.40±1.64	81.62±1.76●	79.97±1.46 ●	80.72±1.78 ●	72.64±1.81 ●	75.54±1.83 ●
zoo	93.86±6.42	94.55±6.54	94.37±6.79	96.73±5.45	92.61±7.33	93.51±7.16
average	85.13±4.52	85.02±4.51	82.82±4.43	84.64±4.39	82.87 ±4.51	83.92±4.41

●, ○ statistically significant degradation or improvement compared with LCLT

Table 19. *t*-test of ACC experimental results on LCLT, NBT, NB, TAN, C4.5 and C4.5-B.

Models	C4.5	C4.5-B	TAN	NB	NBT
C4.5-B	6/27/0				
TAN	8/22/3	3/25/5			
NB	8/13/12	5/15/13	3/19/11		
NBT	7/24/2	5/25/3	3/26/4	11/22/0	
LCLT	8/23/2	4/26/3	6/24/3	11/21/1	3/29/1