Scaling Up the Accuracy of Decision-Tree Classifiers: A Naive-Bayes Combination

Liangxiao Jiang[1] and Chaoqun Li[2]
[1]Department of Computer Science, China University of Geosciences, Wuhan, Hubei 430074, China
[2]Department of Mathematics, China University of Geosciences, Wuhan, Hubei 430074, China
{ljiang, chqli}@cug.edu.cn

***Abstract*— C4.5 and NB are two of the top 10 algorithms in data mining thanks to their simplicity, effectiveness, and efficiency. In order to integrate their advantages, NBTree builds a naive Bayes classifier on each leaf node of the built decision tree. NBTree significantly outperforms C4.5 and NB in terms of classification accuracy. However, it incurs very high time complexity. In this paper, we propose a very simple, effective, and efficient algorithm based on C4.5 and NB. We simply denote it C4.5-NB. Our motivation is to keep the high classification accuracy of NBTree without incurring the high time complexity. In C4.5-NB, C4.5 and NB are built and evaluated independently at the training time, and the class-membership probabilities are weightily averaged according to their classification accuracies on training data at the test time. Empirical studies on a large number of UCI data sets show that it performs as well as NBTree in terms of classification accuracy, but is significantly more efficient than NBTree.**

***Index Terms*— naive Bayes, decision trees, class-membership probabilities, weights, classification, ranking**

I. INTRODUCTION

Classification is one of the fundamental problems in data mining. In classification, the goal is to learn a classifier from a given set of instances with class labels, which correctly assigns a class label to a test instance. The performance of a classifier is usually measured by its classification accuracy. Classification has been extensively studied and various learning algorithms have been developed, such as decision trees and Bayesian networks, that can be categorized into two major approaches: probability-based approaches and decision boundary-based approaches. In this paper, we focus on probability-based approaches.

Assume that A_1, A_2, $\cdots$, A_m are m attributes. A test instance x is represented by a vector $< a_1, a_2, \cdots, a_m >$, where a_i is the value of A_i. We use C to denote the class variable and c to denote its value, and $c(x)$ to denote the class of the test instance x. In a probability-based learning algorithm, a joint probability distribution $P(a_1, a_2, \cdots, a_m, c)$ is learned from the training data, and x is classified into the class $c(x)$ with the maximum posterior class probability $P(c|a_1, a_2, \cdots, a_m)$ (or simply, class probability), as shown below:

$$c(x) = \arg\max_{c \in C} P(c|x) \tag{1}$$

where

$$P(c|x) = P(c|a_1, a_2, \cdots, a_m) \tag{2}$$

C4.5 and NB are two of the top 10 algorithms in data mining [1] thanks to their simplicity, effectiveness, and efficiency. In order to integrate their advantages, Kohavi [2] proposes a hybrid algorithm, simply called NBTree, which builds a naive Bayes classifier on each leaf node of the built decision tree. Just as the author expected, NBTree significantly outperforms C4.5 and NB in terms of classification performance. However, it incurs the high time complexity. In this paper, we propose a very simple, effective, and efficient algorithm based on C4.5 and NB. We simply denote it C4.5-NB. Our motivation is to keep the high classification accuracy of NBTree without incurring the high time complexity.

The rest of the paper is organized as follows. In Section II, III, and IV, we briefly review the naive Bayes classifiers, the decision tree classifiers, and naive Bayes tree classifiers. In Section V, we propose our new algorithm C4.5-NB. In Section VI, we describe the experimental methods and results in detail. In Section VII, we draw conclusions and outline our main directions for future work.

II. C4.5: A DECISION TREE CLASSIFIER

The basic process of building a decision tree can be expressed recursively [3]. First, a best attribute is selected to place at the root node of the tree and create one child node for each possible value of this selected attribute. For each child node, if it isn't a leaf node, the entire process is then repeated recursively only using those training instances that actually reach this node. If it is a leaf node, stop splitting this branch of the tree. Once a decision tree has been built, it classify a test instance by sorting it down the tree from the root node to one leaf node, which provides the classification of this instance via simply voting.

Among decision tree classification algorithms, C4.5 [4] is the most popular one, which is the improved version of ID3 [5]. C4.5 estimates the probability of a test instance x belonging to class c using simple voting at the leaf where x falls into. The detailed equation is:

$$P(c|x)_{C4.5} = \frac{\sum_{i=1}^{k} \delta(c_i, c)}{k} \tag{3}$$

where k is the number of training instances in the leaf node, $\delta(\bullet)$ is a binary function, which is one if its two

doi:10.4304/jcp.6.7.1325-1331

parameters are identical and zero otherwise.

To our knowledge, the recursive partition methods used in C4.5 suffer from the fragmentation problem, and they have been observed to produce poor performance of class probability estimation [6].

Therefore, Improving the class probability estimation performance of the built tree has attracted much attention from researchers. The related methods include the Laplace estimate [6], [7], the m-estimate [3], the similarity-weighted estimate and the naive Bayes-based estimate [8], the kernel-based estimate [9], and so on. Besides, averaging probability estimates from all leaf nodes of the single tree [10] and averaging probability estimates from a bag of trees [7] have achieved significant improvements.

III. NB: A NAIVE BAYES CLASSIFIER

A Bayesian network [11] consists of a structural model and a set of conditional probabilities. The structural model is a directed acyclic graph in which nodes represent attributes and arcs represent attribute dependencies. Attribute dependencies are quantified by conditional probabilities for each node given its parents. Bayesian networks are often used for the classification problems, in which a learner attempts to construct Bayesian network classifiers from a given set of training instances with class labels. Assume that all attributes are fully independent given the class, then the resulting Bayesian network classifiers are called naive Bayesian classifiers (simply NB).

NB uses Equation 4 to estimate the probability of a test instance x belonging to class c.

$$P(c|x)_{NB} = P(c)\prod_{j=1}^{m} P(a_j|c) \tag{4}$$

where m is the number of attributes, a_j is the *j*th attribute value of x, the prior probability $P(c)$ and the conditional probability $P(a_j|c)$ are defined using Equation 5 and Equation 6 respectively.

$$P(c) = \frac{\sum_{i=1}^{n}\delta(c_i,c)+1}{n+n_c} \tag{5}$$

$$P(a_j|c) = \frac{\sum_{i=1}^{n}\delta(a_{ij},a_j)\delta(c_i,c)+1}{\sum_{i=1}^{n}\delta(c_i,c)+n_j} \tag{6}$$

where n is the number of training instances, n_c is the number of classes, c_i is class label of the *i*th training instance, n_j is the number of values of the *j*th attribute, and a_{ij} is the *j*th attribute value of the *i*th training instance.

To construct naive Bayes classifiers, we only need to estimate the probability values of $P(c)$ and $P(a_j|c), j = 1, 2, \cdots, m$, from the training data. So naive Bayes classifiers are very simple, effective, and efficient. However, they require making a strong conditional independence assumption, which often harms its classification performance in many realistic data mining applications.

In order to weaken its conditional independence assumption, many approaches are proposed. The related work can be broadly divided into five main categories [12]: 1) structure extension [13]; 2) attribute weighting [14]; 3) attribute selection [15]; 4) instance weighting [16]; 5) instance selection, also called local learning [17].

IV. NBTREE: A NAIVE BAYES TREE CLASSIFIER

Although the attribute independence assumption of naive Bayes is always violated on the whole training data, it could be expected that the dependencies within the local training data is weaker than that on the whole training data. Thus, NBTree [2] builds a naive Bayes classifier on each leaf node of the built decision tree, which just integrate the advantages of the decision tree classifiers and the naive Bayes classifiers. Simply speaking, it firstly uses decision tree to segment the training data, in which each segment of the training data is represented by a leaf node of tree, and then builds a naive Bayes classifier on each segment.

A fundamental issue in building decision trees is the attribute selection measure at each non-terminal node of the tree. Namely, the utility of each non-terminal node and a split needs to be measured in building decision trees. In NBTree, the utility of a node is measured by computing the 5-fold cross-validation accuracy estimate of using NB at the node. To avoid splits with little value and to address the fragmentation problem confronting C4.5, the author defines a split to be significant if the relative reduction in error is greater than 5% and there are at least 30 instances in the node. All these are totally different from those of C4.5.

NBTree significantly outperforms naive Bayes and C4.5 in terms of classification performance indeed. However, it incurs the high time complexity, because it needs to build and evaluate naive Bayes classifiers again and again in creating a split.

V. C4.5-NB: A VERY SIMPLE, EFFECTIVE, AND EFFICIENT CLASSIFIER BASED ON C4.5 AND NB

In this Section, we propose a very simple, effective, and efficient classifier based on C4.5 and NB. We simply denote it C4.5-NB. Our motivation is to keep the high classification accuracy of NBTree without incurring the high time complexity.

At the training time, C4.5-NB independently builds C4.5 and NB, and then evaluates their classification accuracies (respectively denoted by $ACC_{C4.5}$ and ACC_{NB}) on the training data. At the test time, C4.5-NB estimates the class-membership probabilities using the weighted average of the class-membership probabilities produced by C4.5 and NB respectively, where the related weights are the estimated $ACC_{C4.5}$ and ACC_{NB} respectively. Namely, C4.5-NB uses Equation 7 to estimate the probability of a test instance x belonging to class c.

$$P(c|x)_{C4.5-NB} = \frac{ACC_{C4.5}\times P(c|x)_{C4.5}+ACC_{NB}\times P(c|x)_{NB}}{ACC_{C4.5}+ACC_{NB}} \tag{7}$$

Thus, the whole algorithm of C4.5-NB can be partitioned into a training algorithm ($Training$) and a test algorithm ($Test$). They are depicted as:

Algorithm $Training$ (**D**)
Input : the training data **D**
Output the C4.5 and NB classifiers, $ACC_{C4.5}$, and ACC_{NB}
1) Builds a C4.5 classifier using **D**
2) Evaluates $ACC_{C4.5}$ on **D**
3) Builds a NB classifier using **D**
4) Evaluates ACC_{NB} on **D**
5) Returns the built C4.5 and NB classifiers, $ACC_{C4.5}$, and ACC_{NB}

Algorithm $Test$ (C4.5, NB, $ACC_{C4.5}$, ACC_{NB}, **x**)
Input : the built C4.5 and NB, $ACC_{C4.5}$, ACC_{NB}, and a test instance **x**
Output the class label of **x**
1) For each possible class label c
2) Uses C4.5 to estimate $P(c|x)_{C4.5}$
3) Uses NB to estimate $P(c|x)_{NB}$
4) Estimates $P(c|x)_{C4.5-NB}$
5) Returns c with maximal $P(c|x)_{C4.5-NB}$ as the class label of **x**

In learning C4.5-NB, we need to define the weights of the class-membership probabilities produced by C4.5 and NB. For simplicity, we currently define them as their classification accuracies on the training data, which is called the re-substitution accuracy. Although it is not a reliable predictor of the true accuracy on the new data, it is nevertheless often useful to know.

The time complexity of C4.5-NB is almost equal to the sum of the time complexity of C4.5 and NB. Thinking of the very low time complexity of C4.5 and NB, C4.5-NB also has very low time complexity, and should be widely used for classification in real-world applications.

To some extent, C4.5-NB also is an ensemble classifier, which constructs two classifiers, C4.5 and NB, from training data and then averages their class-membership probabilities when classifying a test instance. However, only simple weighted average, instead of bagging and boosting, is used. Therefore, the efficiency and comprehensibility of it are all better than other ensemble algorithms using bagging [18], [19] and boosting [19], [20] etc.

VI. Experimental Methods and Results

We run our experiments on 36 UCI datasets published on the main web site of Weka platform [21], which represent a wide range of domains and data characteristics. We downloaded these data sets in the format of *arff* from the main web site of Weka. The description of the 36 data sets is shown in Table I.

We conducted empirical experiments to compare C4.5-NB with C4.5, NB, NBtree, Boosted C4.5, Boosted NB, Bagged C4.5, and Bagged NB in terms of classification accuracy and running time. We implement our C4.5-NB in the Weka software and use the implementation of naive Bayes (NiaveBayes), C4.5 (J48 with default setting), NBTree, Boosted C4.5 (AdaBoostM1 with J48 as the basic classifier), Boosted NB (AdaBoostM1 with NiaveBayes as the basic classifier), Bagged C4.5 (Bagging with J48 as the basic classifier), and Bagged NB (Bagging with NiaveBayes as the basic classifier) in Weka software.

TABLE I.
DESCRIPTIONS OF UCI DATA SETS USED IN THE EXPERIMENTS.

Dataset	Size	Attributes	Classes	Miss.	Num.
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-scale	625	5	3	N	Y
breast-cancer	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-statlog	270	14	2	N	Y
hepatitis	155	20	2	Y	Y
hypothyroid	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	20000	17	26	N	Y
lymph	148	19	4	N	Y
mushroom	8124	23	2	Y	N
primary-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	5000	41	3	N	Y
zoo	101	18	7	N	Y

In all experiments, the classification accuracy and running time of an algorithm on each data set are obtained via 10 runs of 10-fold cross-validation. Finally, we conducted a two-tailed t-test with 95% confidence level [22] to compare C4.5-NB with C4.5, NB, NBtree, Boosted C4.5, Boosted NB, Bagged C4.5, and Bagged NB.

Table II shows the detailed classification accuracy of each algorithm on each data set, and the symbols $\circ$ and $\bullet$ in the tables respectively denote statistically significant upgradation or degradation over C4.5-NB with a 95% confidence level. Besides, The averaged classification accuracies and $w/t/l$ values are summarized at the bottom of the table. Each entry $w/t/l$ in the table means that C4.5, NB, NBtree, Boosted C4.5, Boosted NB, Bagged C4.5, and Bagged NB win on w data sets, tie on t data sets, and lose on l data sets, compared to our C4.5-NB.

In the same way, we also observe their running time. The running time of each algorithm is the averaged CPU time in millisecond. Our experiments are performed on a dual-processor 2.93 Ghz Pentium 4 Windows computer with 2Gb RAM. The detailed experimental results are shown in Table III. Please also note that the meaning of the t-test symbols in this table is opposite to that in Table II. Because for running time, a small number is better

TABLE II.
CLASSIFICATION ACCURACY COMPARISONS FOR C4.5-NB VERSUS C4.5, NB, NBTREE, BOOSTED C4.5, BOOSTED NB, BAGGED C4.5, AND BAGGED NB.

Dataset	C4.5-NB	C4.5	NB	NBtree	Boosted C4.5	Boosted NB	Bagged C4.5	Bagged NB
anneal	98.78	98.57	86.59 ●	98.42	99.59 ○	94.05 ●	98.90	86.95 ●
anneal.ORIG	94.00	92.35 ●	75.03 ●	97.13 ○	95.17	80.14 ●	93.97	76.60 ●
audiology	76.32	77.26	72.64	76.82	84.75 ○	79.26	80.84 ○	71.49
autos	81.56	81.77	57.41 ●	77.87	85.46	57.12 ●	82.24	57.12 ●
balance-scale	78.41	77.82	90.53 ○	75.97	78.35	91.68 ○	82.04 ○	90.29 ○
breast-cancer	71.66	74.28	72.70	70.99	66.89	68.68	72.71	73.12
breast-w	95.77	95.01	96.07	96.38	96.08	95.55	96.07	96.04
colic	83.35	85.16	78.70 ●	81.71	81.63	77.62 ●	85.34	78.73 ●
colic.ORIG	66.08	66.31	66.18	69.24	66.31	67.35	66.31	65.96
credit-a	84.12	85.57	77.86 ●	85.46	84.01	81.04 ●	85.71	77.99 ●
credit-g	72.44	71.25	75.16 ○	74.27	70.75	75.14 ○	73.89	75.20 ○
diabetes	75.79	74.49	75.75	75.24	71.69 ●	75.86	75.65	75.64
glass	68.61	67.63	49.45 ●	69.90	75.15 ○	49.63 ●	73.50	49.99 ●
heart-c	80.11	76.94 ●	83.34	80.43	78.79	82.97	78.88	83.37
heart-h	84.06	80.22 ●	83.95	82.26	78.68 ●	84.81	79.93 ●	84.13
heart-statlog	80.26	78.15	83.59	79.26	78.59	82.59	80.59	83.59
hepatitis	80.98	79.22	83.81	80.93	82.38	84.62	80.73	84.13
hypothyroid	99.52	99.54	95.30 ●	99.57	99.65	95.28 ●	99.58	95.46 ●
ionosphere	90.63	89.74	82.17 ●	89.15	93.05	91.06	92.17	82.00 ●
iris	95.53	94.73	95.53	93.80	94.33	94.80	94.67	95.53
kr-vs-kp	99.44	99.44	87.79 ●	97.81 ●	99.59	94.38 ●	99.42	87.69 ●
labor	89.30	78.60 ●	93.57	92.27	87.17	89.60	82.60	93.73
letter	88.31	88.03 ●	64.07 ●	86.60 ●	95.53 ○	64.07 ●	92.73 ○	64.15 ●
lymphography	78.46	75.84	83.13	80.80	80.87	81.27	77.25	83.76
mushroom	100.00	100.00	95.76 ●	100.00	100.00	100.00	100.00	95.77 ●
primary-tumor	45.49	41.39 ●	49.71 ○	47.50	41.65 ●	49.71 ○	43.90	49.35
segment	96.36	96.79 ○	80.17 ●	95.28 ●	98.12 ○	80.17 ●	97.39 ○	80.28 ●
sick	98.45	98.72	92.75 ●	97.82 ●	98.99 ○	93.60 ●	98.81 ○	92.72 ●
sonar	74.49	73.61	67.71	77.07	79.13	80.77	78.51	68.21
soybean	93.54	91.78 ●	92.94	92.87	92.83	92.98	92.78	92.78
splice	95.83	94.03 ●	95.41	95.40	93.59 ●	93.68 ●	94.34 ●	95.43
vehicle	72.41	72.28	44.68 ●	70.99	75.59 ○	44.68 ●	74.48	45.58 ●
vote	96.32	96.57	90.02 ●	95.03	95.51	95.01	96.27	90.02 ●
vowel	80.78	80.20	62.90 ●	92.34 ○	92.88 ○	80.14	90.28 ○	63.43 ●
waveform-5000	76.44	75.25 ●	80.01 ○	80.16 ○	81.40 ○	80.01 ○	81.79 ○	79.98 ○
zoo	95.48	92.61	94.97	94.44	95.18	97.23	93.21	95.07
Average	84.42	83.37	79.37	84.76	85.26	81.29	85.21	79.48
w/t/l	-	1/26/9	4/16/16	3/29/4	9/23/4	4/19/13	7/27/2	3/17/16

than a large number, which is opposite to classification accuracy. So ● represents statistically significant better than C4.5-NB in terms of running time and ○ represents worse.

From our experiments, we can see that C4.5-NB performs as well as NBTree in classification accuracy, but is significantly more efficient than NBTree. Now, we summarize some highlights briefly as follows:

1) C4.5-NB significantly outperforms C4.5. C4.5-NB wins C4.5 on 9 data sets, whereas C4.5 only wins C4.5-NB on 1 data sets. Besides, the averaged classification accuracy of C4.5-NB (84.42%) is much higher than that of C4.5 (83.37%).
2) C4.5-NB significantly outperforms NB. C4.5-NB wins NB on 16 data sets, whereas NB only wins C4.5-NB on 4 data sets. Besides, the averaged classification accuracy of C4.5-NB (84.42%) is much higher than that of NB (79.37%).
3) C4.5-NB performs as well as NBTree in classification accuracy (4 wins and 3 losses), but is significantly more efficient than NBTree (34 wins and 0 losses).
4) In terms of classification accuracy, C4.5-NB significantly outperforms boosted NB (13 wins and 4 losses) and bagged NB (16 wins and 3 losses), and almost ties boosted C4.5 (4 wins and 9 losses) and bagged C4.5 (2 wins and 7 losses).
5) In terms of running time, C4.5-NB significantly outperforms boosted C4.5 (34 wins and 1 losses), boosted NB (32 wins and 1 losses) and bagged C4.5 (34 wins and 0 losses), and almost ties bagged NB (6 wins and 7 losses).

In our another group of experiments, we investigate the ranking performance of our C4.5-NB in term of the area under the Receiver Operating Characteristics curve, simply AUC [23]. The experimental results in Table IV show that its ranking performance almost ties NBTree (6 wins and 1 losses) and boosted C4.5 (9 wins and 7 losses) and significantly outperforms all the other algorithms used to compare.

Besides, we design a group of experiments to compare our proposed method with following two methods [24], [25]: 1) Simple voting using average of probabilities of NB and C4.5, simply denoted by *Averaged*. 2) Simple voting using product of probabilities of NB and C4.5, simply denoted by *Producted*. The experimental results

TABLE III.
RUNNING TIME COMPARISONS FOR C4.5-NB VERSUS C4.5, NB, NBTREE, BOOSTED C4.5, BOOSTED NB, BAGGED C4.5, AND BAGGED NB.

Dataset	C4.5-NB	C4.5	NB	NBtree	Boosted C4.5	Boosted NB	Bagged C4.5	Bagged NB
anneal	94.87	51.41 •	9.69 •	6453.88 ◦	389.69 ◦	734.22 ◦	468.61 ◦	91.05
anneal.ORIG	129.48	90.47 •	9.83 •	7271.88 ◦	627.63 ◦	643.82 ◦	894.49 ◦	90.97 •
audiology	34.36	13.92 •	4.08 •	20351.01 ◦	202.97 ◦	354.22 ◦	137.67 ◦	31.78
autos	31.57	14.05 •	3.14 •	2219.03 ◦	167.84 ◦	146.24 ◦	138.43 ◦	34.38
balance-scale	19.83	12.97	1.88 •	285.00 ◦	178.61 ◦	119.48 ◦	117.84 ◦	20.21
breast-cancer	2.97	2.33	0.31	362.01 ◦	34.99 ◦	12.05 ◦	24.22 ◦	3.27
breast-w	25.75	12.81 •	4.53 •	328.76 ◦	201.41 ◦	166.45 ◦	117.18 ◦	42.69 ◦
colic	20.47	15.47	1.86 •	3031.58 ◦	257.49 ◦	75.13 ◦	144.54 ◦	20.36
colic.ORIG	11.13	5.94	1.56 •	8774.37 ◦	22.17 ◦	88.24 ◦	61.71 ◦	22.22 ◦
credit-a	33.12	22.19 •	2.34 •	1395.45 ◦	295.47 ◦	110.42 ◦	197.32 ◦	34.84
credit-g	58.15	40.79 •	5.95 •	1476.26 ◦	488.26 ◦	270.30 ◦	375.94 ◦	63.41
diabetes	38.09	26.70 •	4.04 •	288.60 ◦	416.85 ◦	178.02 ◦	321.06 ◦	42.83
glass	25.14	15.15	2.05 •	494.36 ◦	164.18 ◦	47.52 ◦	132.07 ◦	20.34
heart-c	16.88	9.07	1.90 •	776.74 ◦	125.17 ◦	137.47 ◦	84.82 ◦	19.43
heart-h	16.86	9.85	1.71 •	550.45 ◦	208.28 ◦	100.07 ◦	89.70 ◦	16.23
heart-statlog	17.52	11.55	2.03 •	516.11 ◦	145.49 ◦	107.92 ◦	103.75 ◦	24.07
hepatitis	6.12	5.32	0.78	799.23 ◦	67.36 ◦	32.14 ◦	45.15 ◦	7.66
hypothyroid	170.90	69.39 •	29.19 •	13648.25 ◦	1204.20 ◦	1275.95 ◦	683.74 ◦	288.61 ◦
ionosphere	91.07	70.93 •	7.81 •	5378.47 ◦	664.39 ◦	367.61 ◦	577.97 ◦	76.66 •
iris	3.43	0.94	0.47	62.66 ◦	15.31 ◦	34.33 ◦	11.09	4.51
kr-vs-kp	83.94	58.15 •	12.82 •	14048.86 ◦	1177.28 ◦	432.10 ◦	567.03 ◦	104.09 ◦
labor	2.19	0.93	0.16	266.12 ◦	15.16 ◦	10.63 ◦	10.14	2.96
letter	11697.27	5366.53 •	835.33 •	375519.68 ◦	57571.08 ◦	59671.13 ◦	42855.57 ◦	8467.89 •
lymphography	4.72	2.67	0.62	982.06 ◦	34.06 ◦	45.43 ◦	25.17 ◦	7.20
mushroom	70.26	30.18 •	16.91 •	4244.40 ◦	47.51 •	756.86 ◦	286.88 ◦	177.04 ◦
primary-tumor	20.17	11.38	1.59 •	1727.95 ◦	83.03 ◦	26.86	116.75 ◦	10.78
segment	515.22	243.44 •	52.11 •	19271.12 ◦	2916.71 ◦	1293.38 ◦	2233.40 ◦	520.45
sick	219.26	157.82 •	23.94 •	21547.88 ◦	2416.82 ◦	855.01 ◦	1199.52 ◦	247.83 ◦
sonar	82.07	61.09 •	7.20 •	4870.48 ◦	656.26 ◦	367.56 ◦	540.45 ◦	80.59
soybean	55.94	29.82 •	4.23 •	22639.24 ◦	369.70 ◦	419.90 ◦	269.74 ◦	42.42 •
splice	212.25	161.86 •	17.39 •	26710.00	342.84 ◦	857.37 ◦	1628.01 ◦	185.11 •
vehicle	135.46	-1638.15	12.35 •	6526.88 ◦	1047.53 ◦	49.03 •	898.29 ◦	125.06
vote	6.55	4.39	1.10	-474.36	68.73 ◦	30.94 ◦	39.85 ◦	7.49
vowel	274.52	178.89 •	13.28 •	5898.73 ◦	1968.59 ◦	1571.30 ◦	1573.05 ◦	139.11 •
waveform-5000	2476.40	1907.95 •	170.32 •	47266.21 ◦	23350.11 ◦	7053.99	17715.85 ◦	1705.89 •
zoo	2.96	0.78	0.47	831.21 ◦	10.62	8.74	13.12 ◦	0.78
Average	464.08	196.64	35.14	17398.35	2720.94	2179.22	2075.00	355.01
w/t/l	-	0/16/20	0/7/29	34/2/0	34/1/1	32/3/1	34/2/0	6/23/7

in Table V and Table VI show that:

1) Our proposed method significantly outperforms simple voting using average of probabilities of NB and C4.5 in terms of classification accuracy (8 wins and 1 losses).
2) Our proposed method significantly outperforms simple voting using product of probabilities of NB and C4.5 in terms of AUC (19 wins and 0 losses).

VII. CONCLUSIONS AND FUTURE WORK

Due to the simplicity, effectiveness, and efficiency, C4.5 and NB are two very important algorithms for addressing the classification problems. In this paper, we propose a very simple, effective, and efficient algorithm based on C4.5 and NB. We simply denote it C4.5-NB. In C4.5-NB, C4.5 and NB are built and evaluated independently at the training time, and the class-membership probabilities are weightily averaged according to their classification accuracies on training data at the test time. The experimental results on a large number of UCI data sets show that it performs as well as NBTree in classification accuracy, but is significantly more efficient than NBTree.

C4.5-NB needs to define the weights of the class-membership probabilities produced by C4.5 and NB. In our current version, we simply define them as their classification accuracies on the training data. Therefore, we believe that the use of more sophisticated methods, such as the accuracy estimates based on k-fold cross-validation or leave-one-out, could improve the performance of the current C4.5-NB and make its advantage stronger. This is the main research direction for our future work.

ACKNOWLEDGEMENTS

This paper is an extended version of ISICA 2008 conference paper [26]. We thank anonymous reviewers for their valuable comments and suggestions. The work was supported by the National Natural Science Foundation of China (No. 60905033), the Provincial Natural Science Foundation of Hubei (No. 2009CDB139), and the Fundamental Research Funds for the Central Universities (No. CUG090109).

REFERENCES

[1] K. V. Q. J. R. e. a. Wu, X., "Top 10 algorithms in data mining," *Knowledge and Information Systems*, vol. 14, pp. 1–37, 2008.

TABLE IV.
AUC COMPARISONS FOR C4.5-NB VERSUS C4.5, NB, NBTREE, BOOSTED C4.5, BOOSTED NB, BAGGED C4.5, AND BAGGED NB.

Dataset	C4.5-NB	C4.5	NB	NBtree	Boosted C4.5	Boosted NB	Bagged C4.5	Bagged NB
anneal	96.21	86.20 •	96.05	95.64	93.28	94.06	88.21 •	95.99
anneal.ORIG	95.27	90.95 •	95.35	95.27	95.03	88.53 •	92.45 •	95.31
audiology	71.21	69.43 •	71.12	71.07	70.94	70.90	69.93 •	71.11
autos	91.88	84.68 •	87.54 •	93.34	95.16 ○	84.91 •	90.97	87.86 •
balance-scale	82.56	68.61 •	89.30 ○	72.43 •	87.05 ○	96.44 ○	82.88	89.41 ○
breast-cancer	68.10	56.67 •	68.44	64.39	61.24	62.39	64.17	69.04
breast-w	98.37	95.64 •	98.77	98.49	98.65	97.91	98.71	98.91
colic	86.46	82.87	82.91 •	85.12	85.60	82.91	85.94	83.14 •
colic.ORIG	77.84	50.00 •	77.84	68.87 •	71.40 •	75.95	68.06 •	77.50
credit-a	91.57	88.59 •	89.48 •	91.53	90.48	87.24 •	92.56	89.47 •
credit-g	76.17	64.78 •	78.22 ○	75.90	71.45 •	75.41	73.62	78.19
diabetes	81.63	75.18 •	81.36	79.95	76.77 •	80.34	81.02	81.49
glass	88.38	79.24 •	84.65 •	86.71	89.25	76.36 •	86.03	84.47 •
heart-c	83.85	82.73 •	84.01	83.74	83.57	83.64	83.63	84.00
heart-h	83.90	82.57 •	84.02	83.86	83.61	83.83	83.83	84.02
heart-statlog	88.86	79.59 •	89.49	83.34	84.97	86.50	86.98	89.63
hepatitis	84.01	65.93 •	85.03	81.32	81.92	75.37	80.25	87.00
hypothyroid	98.77	98.27	94.06 •	98.17	98.39	91.38 •	98.30	94.37 •
ionosphere	94.48	88.07 •	93.42	91.56	96.75	94.89	96.68	93.60
iris	99.33	87.80 •	99.32	95.98 •	96.15 •	98.35	90.87 •	99.30
kr-vs-kp	99.82	99.82	95.16 •	99.44	99.95 ○	98.76 •	99.92 ○	95.17 •
labor	94.04	62.54 •	95.67	95.67	92.88	89.50	87.42	96.92
letter	98.59	94.70 •	95.50 •	97.84 •	99.86 ○	84.60 •	99.27 ○	95.56 •
lymphography	88.92	81.85 •	89.70	88.77	88.81	87.71	87.18	89.40
mushroom	100.00	99.88 •	99.81 •	100.00	99.88 •	100.00	99.88 •	99.81 •
primary-tumor	78.91	73.33 •	79.31	78.70	75.07 •	76.96	75.17 •	79.36
segment	99.45	97.40 •	97.81 •	98.55 •	99.86 ○	93.93 •	98.73 •	97.90 •
sick	98.05	94.96 •	92.30 •	93.56 •	98.98	91.40 •	99.22	92.50 •
sonar	81.77	73.53 •	78.96	81.87	87.04	87.03	86.37	78.80
soybean	99.80	83.97 •	99.74	99.53	95.89 •	93.54 •	85.77 •	99.75
splice	99.40	96.80 •	99.45	99.43	96.28 •	98.64 •	98.47 •	99.45
vehicle	85.52	81.61 •	74.42 •	86.64	91.39 ○	74.42 •	91.52 ○	75.07 •
vote	98.18	97.73	97.18	98.66	98.68	96.90	98.39	97.22
vowel	97.93	88.35 •	95.15 •	99.01 ○	99.62 ○	97.31	97.30 •	95.36 •
waveform-5000	93.41	82.23 •	95.66 ○	93.40	93.85	88.40 •	94.72 ○	95.68 ○
zoo	89.48	88.33 •	89.48	89.48	88.34 •	89.17	88.36 •	89.48
Average	90.06	82.63	89.05	88.81	89.39	87.10	88.41	89.20
w/t/l	-	0/4/32	3/21/12	1/29/6	7/20/9	1/22/13	4/20/12	2/22/12

[2] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid," ser. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. AAAI Press, 1996, pp. 202–207.

[3] T. M. Mitchell, *Machine Learning*, 1st ed. McGraw-Hill, 1997.

[4] J. R. Quinlan, *C4.5: Programs for Machine Learning*, 1st ed. San Mateo, CA: Morgan Kaufmann, 1993.

[5] Q. J. Ross, "Induction of decision trees," *Machine Learning*, vol. 1, pp. 81–106, 1986.

[6] F. T. Provost, F. and R. Kohavi, "The case against accuracy estimation for comparing induction algorithms," ser. Proceedings of the 15th International Conference on Machine Learning, 1998, pp. 445–453.

[7] F. Provost and P. Domingos, "Tree induction for probability-based ranking," *Machine Learning*, vol. 52, pp. 199–215, 2003.

[8] L. C. Jiang, L. and Z. Cai, "Learning decision tree for ranking," *Knowledge and Information Systems*, vol. 20, pp. 123–135, 2009.

[9] G. A. G. E. Smyth, P. and U. M. Fayyad, "Retrofitting decision tree classifiers using kernel density estimation," ser. Proceedings of the 12th International Conference on Machine Learning. Morgan Kaufmann, 1995, pp. 506–514.

[10] C. Ling and R. Yan, "Decision tree with better ranking," ser. Proceedings of the Twentieth International Conference on Machine Learning. AAAI Press, 2003, pp. 480–487.

[11] J. Pearl, *Probabilistic Reasoning in Intelligent Systems*. San Francisco: Morgan Kaufmann, 1988.

[12] W. D. C. Z. Jiang, L. and X. Yan, "Survey of improving naive bayes for classification," ser. Proceedings of the 3rd International Conference on Advanced Data Mining and Applications. Springer Press, 2007, pp. 134–145.

[13] N. Friedman, D. Geiger, and M. Goldszmidt, "Bayesian network classifiers," *Machine Learning*, vol. 29, pp. 131–163, 1997.

[14] M. A. Hall, "A decision tree-based attribute weighting filter for naive bayes," *Knowledge-Based Systems*, vol. 20, pp. 120–126, 2007.

[15] P. Langley and S. Sage, "Induction of selective bayesian classifiers," ser. Proceedings of the 12th Conference on Uncertainty in Artificial Intelligence, 1994, pp. 339–406.

[16] C. Z. Jiang, L. and D. Wang, "Improving naive bayes for classification," *International Journal of Computers and Applications*, vol. 32, pp. 328–332, 2010.

[17] H. M. Frank, E. and B. Pfahringer, "Locally weighted naive bayes," ser. Proceedings of the Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, 2003, pp. 249–256.

[18] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.

[19] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting and variants," *Machine Learning*, vol. 36, pp. 105–142, 1999.

[20] R. E. S. Y. Freund, "Experiments with a new boosting algorithm," ser. Proceedings of the 13th International Con-

TABLE V.
CLASSIFICATION ACCURACY COMPARISONS FOR C4.5-NB VERSUS *Averaged* AND *Producted*.

Dataset	C4.5-NB	*Averaged*	*Producted*
anneal	98.78	98.70	98.95
anneal.ORIG	94.00	90.12 ●	92.62 ●
audiology	76.32	75.91	76.43
autos	81.56	80.44	80.75
balance-scale	78.41	78.41	78.40
breast-cancer	71.66	71.62	71.62
breast-w	95.77	95.74	95.74
colic	83.35	83.07	83.07
colic.ORIG	66.08	66.13	66.13
credit-a	84.12	81.55 ●	81.55 ●
credit-g	72.44	72.51	72.51
diabetes	75.79	75.90	75.90
glass	68.61	65.94	63.51
heart-c	80.11	80.27	80.27
heart-h	84.06	83.95	83.95
heart-statlog	80.26	80.89	80.89
hepatitis	80.98	81.98	81.47
hypothyroid	99.52	99.20 ●	99.41
ionosphere	90.63	88.04	87.92
iris	95.53	95.53	95.00
kr-vs-kp	99.44	99.38	99.38
labor	89.30	86.87	87.03
letter	88.31	87.71 ●	87.09 ●
lymphography	78.46	78.73	78.68
mushroom	100.00	100.00	100.00
primary-tumor	45.49	45.73	45.11
segment	96.36	92.95 ●	93.60 ●
sick	98.45	98.44	98.45
sonar	74.49	74.77	74.77
soybean	93.54	93.56	93.57
splice	95.83	95.63 ●	95.62
vehicle	72.41	70.67 ●	71.90
vote	96.32	92.73 ●	92.73 ●
vowel	80.78	81.03	81.49
waveform-5000	76.44	77.33 ○	77.13 ○
zoo	95.48	95.18	95.18
Average	84.42	83.79	83.83
w/t/l	-	1/27/8	1/30/5

TABLE VI.
AUC COMPARISONS FOR C4.5-NB VERSUS *Averaged* AND *Producted*.

Dataset	C4.5-NB	*Averaged*	*Producted*
anneal	96.21	96.21	86.40 ●
anneal.ORIG	95.27	95.26	92.24 ●
audiology	71.21	71.21	69.76 ●
autos	91.88	91.75	85.14 ●
balance-scale	82.56	82.53	77.46 ●
breast-cancer	68.10	68.09	68.40
breast-w	98.37	98.38	98.52
colic	86.46	86.38	86.02
colic.ORIG	77.84	77.84	77.84
credit-a	91.57	91.32 ●	91.38
credit-g	76.17	76.36 ○	76.31
diabetes	81.63	81.70	81.06
glass	88.38	87.72	79.34 ●
heart-c	83.85	83.86	83.72
heart-h	83.90	83.90	83.98
heart-statlog	88.86	89.00	87.29
hepatitis	84.01	84.06	83.57
hypothyroid	98.77	98.74	98.48 ●
ionosphere	94.48	94.46	92.70
iris	99.33	99.33	91.50 ●
kr-vs-kp	99.82	99.82	99.86
labor	94.04	93.87	89.54
letter	98.59	98.57 ●	94.77 ●
lymphography	88.92	88.94	85.71
mushroom	100.00	100.00	99.88 ●
primary-tumor	78.91	78.90	74.41 ●
segment	99.45	99.38 ●	97.36 ●
sick	98.05	97.99 ●	97.90 ●
sonar	81.77	82.45	80.14
soybean	99.80	99.80	84.85 ●
splice	99.40	99.39	98.07 ●
vehicle	85.52	85.31	81.10 ●
vote	98.18	98.07	97.52
vowel	97.93	97.92	88.91 ●
waveform-5000	93.41	93.66 ○	87.02 ●
zoo	89.48	89.48	88.33 ●
Average	90.06	90.05	87.13
w/t/l	-	2/30/4	0/17/19

ference on Machine Learning. Morgan Kaufmann, 1996, pp. 148–156.

[21] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. San Francisco: Morgan Kaufmann, 2005.

[22] C. Nadeau and Y. Bengio, "Inference for the generalization error," *Machine Learning*, vol. 52, pp. 239–281, 2003.

[23] D. J. Hand and R. J. Till, "A simple generalisation of the area under the ROC curve for multiple class classification problems," *Machine Learning*, vol. 45, pp. 171–186, 2001.

[24] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. John Wiley and Sons, Inc., 2004.

[25] R. P. D. J. M. J. Kittler, M. Hatef, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, pp. 226–239, 1998.

[26] J. W. L. Jiang, C. Li and J. Zhu., "A combined classification algorithm based on c4.5 and nb," ser. Proceedings of the 3rd International Symposium on Intelligent Computation and its Applications. Springer Press, 2008, pp. 350–359.

BIOGRAPHIES

Liangxiao Jiang received his PhD degree from China University of Geosciences. Currently, he is an associate professor in Department of Computer Science at China University of Geosciences. His research interests include data mining and machine learning.

Chaoqun Li is currently a Ph.D. candidate at China University of Geosciences. Her research interests include data mining and machine learning.