An Empirical Study on Class Probability Estimates in Decision Tree Learning

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*— Decision tree is one of the most effective and widely used models for classification and ranking and has received a great deal of attention from researchers in the domain of data mining and machine learning. A critical problem in decision tree learning is how to estimate the class-membership probabilities from decision trees. In this paper, we firstly survey all kinds of class probability estimation methods, mainly include the maximum-likelihood estimate, the Laplace estimate, the m-estimate, the similarity-weighted estimate, the naive Bayes-based estimate, and so on. Then, we provide an empirical study on the classification and ranking performance of the resulting decision trees using different class probability estimation methods. The experimental results based on a large number of UCI data sets verify our conclusions.**

***Index Terms*— decision tree learning; probability estimation tree; class probability estimation; classification; ranking.**

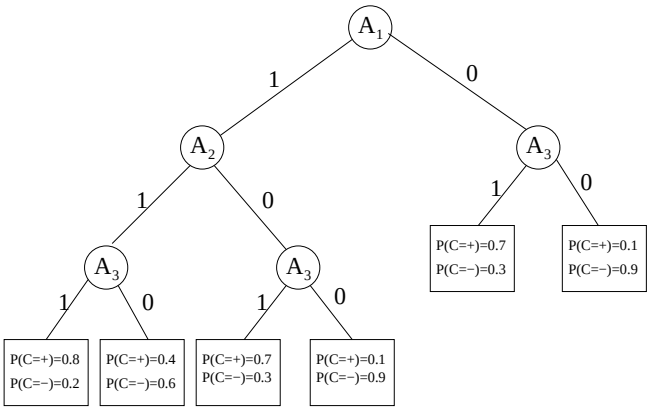


Figure 1. An example of PETs

I. INTRODUCTION

Due to being efficient, effective, robust to noisy data, and capable of learning disjunctive expressions, decision tree learning has received a great deal of attention from researchers in the domain of data mining and machine learning. It uses a decision tree to represent a discrete-valued target function [1] and has been successfully applied to a broad range of tasks such as learning to classify medical patients by their disease and rank loan applicants by their likelihood of defaulting on payments.

At first, let's rewrite the basic decision tree learning algorithm as:

Algorithm: basic decision tree learning algorithm
Input: a training instances set $\mathbf{D}$
Output: built decision tree $\mathbf{T}$

1) If all instances in $\mathbf{D}$ have the same class label, or have the same attribute values except for the class label, or $\mathbf{D}$ is empty, then creates a leaf node using $\mathbf{D}$.
2) Else a best attribute is selected to partition $\mathbf{D}$ into several smaller subsets and a child node is created for each subsets.
3) The algorithm is then recursively applied to each child node till all child nodes are leaf nodes.
4) Returns the built decision tree $\mathbf{T}$.

Once a decision tree has been built, it predicts (classifies or ranks) an unseen instance by sorting it down the tree from the root node to some leaf node, and then using the training instances that fall into this leaf node to estimate its class-membership probabilities. Such learned trees are generally called probability estimation trees (simply PETs).

Figure 1 shows an example of PETs. The target function represented by it only has two classes: the positive class $+$ and the negative class $-$. Estimating class-membership probabilities from small instance sets is a well-studied statistical problem, and a thorough study of what are the best methods (and why) for PETs is a useful contribution to machine-learning research [2]. In this paper, we focus our attention to discuss how to learn this kind of PETs, namely how to estimate the class-membership probabilities from built decision trees.

The rest of the paper is organized as follows. Some alternative methods for estimating the class-membership probabilities are summarized in Section II. the experimental methodology and results are given in Section III. In Section IV, we draw conclusions.

II. METHODS FOR CLASS PROBABILITY ESTIMATES

There exist many methods for estimating the class-membership probabilities from probability estimation trees. For example, the maximum-likelihood estimate, the Laplace estimate, the m-estimate, similarity-weighted estimate, the naive Bayes-based estimate, and so on.

doi:10.4304/jsw.6.7.1368-1373

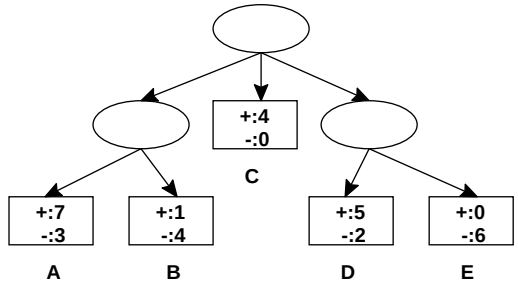


Figure 2. An example of a decision tree for class probability estimation

The maximum-likelihood estimate calculates the probability $P(c|x)$ that an unseen instance x falling into the leaf node L belongs to the class c as:

$$P(c|x) = \frac{\sum_{i=1}^{n} \delta(c_i, c)}{n} \tag{1}$$

where n is the number of instances in the leaf node L, c_i is class label of the *i*th training instance in the leaf node L, and $\delta(c_i, c)$ is one if $c_i = c$ and zero otherwise. For example, in Figure 2, if an unseen instance x falls into the leaf node A, according to the maximum-likelihood estimate, the probability $P(+|x)$ that x belongs to the class $+$ is $\frac{7}{10} = 0.7$.

Obviously, the maximum-likelihood estimate is a purely frequency-based estimate. Thus, a potential problem with it is the zero-probability problem and the one-probability problem. For example, in Figure 2, the leaf node E comprises only 6 training instances, and all of them are of the negative class. According to the maximum-likelihood estimate, if x falls into the leaf node E, then the probability that x belongs to the class $+$ and the class $-$ respectively is 0 and 1. According to the observation by Provost and Domingos [2], such extreme probabilities maybe not reasonable, after all the estimate only from 6 instances is not enough evidence for such a strong statement.

In order to address the problem confronting the maximum-likelihood estimate, Laplace correction is used to smooth the estimated probabilities, and the resulting estimate is the so-called Laplace estimate. The Laplace estimate can be viewed as a combination of the maximum-likelihood estimate and an uniform prior probability. Thus, the Laplace estimate calculates the probability $P(c|x)$ that x falling into the leaf node L belongs to the class c as:

$$P(c|x) = \frac{\sum_{i=1}^{n} \delta(c_i, c) + 1}{n + n_c} \tag{2}$$

where n_c is the number of classes. For the same example above, according to the Laplace estimate, the probability $P(+|x)$ that x belongs to the class $+$ is $\frac{7+1}{10+2} = \frac{8}{12} = 0.67$.

The m-estimate [1] is another method to estimate probability, which has already been applied to improve the class probability estimation of Bayesian classifiers successfully [3]. In this paper, we try to investigate its application in decision tree learning. The m-estimate can be comprehend as augmenting the actual observations by an additional m virtual instances distributed according to p. Thus, the m-estimate calculates the probability $P(c|x)$ that x falling into the leaf node L belongs to the class c as:

$$P(c|x) = \frac{\sum_{i=1}^{n} \delta(c_i, c) + mp}{n + m} \tag{3}$$

where m and p are two parameters. The parameter p is the prior estimate of the probability we wish to determine and the parameter m is a constant called the *equivalent instance size*, which determines how heavily to weight p relative to the observed data. Obviously, the maximum-likelihood estimate is a special example of the m-estimate when the parameter m is 0, and the Laplace estimate estimate is a special example of the m-estimate when the parameter m is n_c and the parameter p is $1/n_c$.

In our implementation, we set the parameter p to an uniform distribution $1/n_c$ and set the parameter m to 1. So, the resulting estimate is:

$$P(c|x) = \frac{\sum_{i=1}^{n} \delta(c_i, c) + 1/n_c}{n + 1} \tag{4}$$

For the same example above, according to the m-estimate we implemented, the probability $P(+|x)$ that x belongs to the class $+$ is $\frac{7+1/2}{10+1} = \frac{7.5}{11} = 0.68$.

From all of above three methods, we can see that any unseen instance falling into a particular leaf node will receive the same class-membership probabilities because only the class variable is used for estimating class-membership probabilities. To address this problem, our previous work [4] pays attention to estimating probabilities from the moderate-size leaf nodes (the least number of instances in leaf nodes is set to 30.) and presents the similarity-weighted estimate and the naive Bayes-based estimate, which take some attribute variables into the class probability estimation.

The similarity-weighted estimate calculates the probability $P(c|x)$ that x falling into the leaf node L belongs to the class c as:

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

where $s(x_i, x)$ is the similarity between x and x_i (the *i*th training instance in the leaf node L), which can be defined as:

$$s(x_i, x) = \sum_{j=1}^{m} \delta(a_{ij}, a_j) \tag{6}$$

where m is the number of attributes, a_{ij} is the *j*th attribute value of x_i, and a_j is the *j*th attribute value of x. In our previous paper [4], we assume that all attribute values are nominal. Thus, the similarity is a function that simply counts the number of identical attribute values of x_i and x.

Different from the similarity-weighted estimate, the naive Bayes-based estimate deploys a naive Bayes on the leaf node L where x falls into. So, it calculates the

probability $P(c|x)$ that x belongs to the class c as:

$$P(c|x) = P(c)\prod_{j=1}^{m} P(a_j|c) \tag{7}$$

where the prior probability $P(c)$ and the conditional probability $P(a_j|c)$ can be defined as:

$$P(c) = \frac{\sum_{i=1}^{n}\delta(c_i,c)+1}{n+n_c} \tag{8}$$

$$P(a_j|c) = \frac{\sum_{i=1}^{n}\delta(a_{ij},a_j)\delta(c_i,c)+1}{\sum_{i=1}^{n}\delta(c_i,c)+n_j} \tag{9}$$

where n_j is the number of values of the *j*th attribute.

Beside, Smyth etc. [5] focus their attention to estimate probabilities from the larger leaf nodes and present a kernel-based method. It places a kernel-based probability density estimator at each leaf node of the decision tree. It estimates the probability $P(c|x)$ that x belongs to the class c as:

$$P(c|x) = \frac{f(x|c)P(c)}{\sum_{c=1}^{n_c} f(x|c)P(c)} \tag{10}$$

where $P(c)$ is the prior probability of class c, which can be estimated from the data in the usual fashion. $f(x|c)$ is the density estimate for the data from class c, which can be estimated using the methods described in the paper by Smyth etc. [5].

Ling [6] single out another new method for estimating class-membership probabilities. Instead of estimating the probabilities at the single leaf node where an unseen instance falls into, it averages probability estimates from all leaf nodes of the tree. The contribution of each leaf node in the average is determined by the deviation in attribute values from the root node to the leaf node. The detailed Equation for estimation is:

$$P(c|x) = \frac{\sum P_i(c)\cdot s^j}{\sum s^j} \tag{11}$$

where $P_i(c)$ is the probability of class c in the ith leaf node, j is the number of split attribute values in the path from the ith leaf node to the root node that are different from the attribute values of x, s is the parameter called confusion factor, which is set to 0.2 in Ling and Yan's paper [6].

For example, in Figure 1, a test instance x with attribute values $A_1 = 0$, $A_2 = 1$, and $A_3 = 0$, will fall into the rightmost leaf node. According to Ling and Yan's conclusion, the probability that x belongs to class $+$ is:

$$P(+|x) = \frac{0.8\cdot 0.2^2+0.4\cdot 0.2^1+0.7\cdot 0.2^3+0.1\cdot 0.2^2+0.7\cdot 0.2^1+0.1\cdot 0.2^0}{0.2^2+0.2^1+0.2^3+0.2^2+0.2^1+0.2^0} = 0.243 \tag{12}$$

So far, we only discuss the methods of estimating class-membership probabilities from a single tree. Recently, averaging multiple decision trees to produce probability estimates has received a great deal of attention. For example, a bagging [7], [8] of C4.4, simply bagged C4.4 [2], has been shown to significantly outperform single decision tree with surprising consistency. Moreover, according to the conclusions drawn by Provost and Domingos [2], once bagging is used, whether or not pruning and the Laplace correction are used makes little difference. Despite its effectiveness, bagging incurs the high time complexity. Besides, when it is used, the comprehensibility of a single tree is lost. Thus, when high-accuracy prediction is solely required, bagging should be used clearly. When comprehensibility and/or computational cost are also important, a single tree should be firstly considered.

III. Experimental Methodology and Results

We run our experiments under the framework of Weka [9] to study the effectiveness of all kinds of class probability estimation methods. In our experiments, we implemented the unpruned decision trees with different class probability estimation methods. Besides, the heuristic measure is adopted [10], which firstly calculates the information gain of each attribute, and then applies the gain ratio measure only to those attributes with information gain value above the average. Now, we introduce established class probability estimation methods and their abbreviations used in our implements and experiments.

1) LE: the Laplace estimate defined by Equation 2. The resulting decision tree algorithm actually is C4.4 [2].
2) ME: the m-estimate defined by Equation 4. The resulting decision tree algorithm actually is C4.4 [2] but with the m-estimate.
3) SWE: the similarity-weighted estimate defined by Equation 5. The resulting decision tree algorithm actually is SWC4.4 [4].
4) NBE: the naive Bayes-based estimate defined by Equation 7. The resulting decision tree algorithm actually is NBC4.4 [4].
5) LE-Bagging: a bagging of the Laplace estimate defined by Equation 2. We use the implementation of Bagging in Weka software with C4.4 as the basic classifier. The resulting decision tree algorithm actually is bagged C4.4 [2].

We run our experiments on 36 UCI datasets published on the main web site of Weka platform [9], 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. In our experiments, numeric values are discretized using ten-bin discretization implemented in Weka, and missing values are also processed using the mechanism in Weka, which replaces all missing values with the modes and means from the training instances. Besides, three useless attributes: the attribute "Hospital Number" in the data set "colic.ORIG", the attribute "instance name" in the data set "splice" and the attribute "animal" in the data set "zoo" are removed by using the unsupervised filter named *Remove* in Weka.

We conducted empirical experiments to compare decision trees resulted from all kinds of class probability estimation methods in terms of classification (measured by classification accuracy) and ranking (measured by

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

No.	Dataset	Instance number	Attribute number	Class number	Missing value	Numeric value
1	anneal	898	39	6	Y	Y
2	anneal.ORIG	898	39	6	Y	Y
3	audiology	226	70	24	Y	N
4	autos	205	26	7	Y	Y
5	balance-scale	625	5	3	N	Y
6	breast-cancer	286	10	2	Y	N
7	breast-w	699	10	2	Y	N
8	colic	368	23	2	Y	Y
9	colic.ORIG	368	28	2	Y	Y
10	credit-a	690	16	2	Y	Y
11	credit-g	1000	21	2	N	Y
12	diabetes	768	9	2	N	Y
13	Glass	214	10	7	N	Y
14	heart-c	303	14	5	Y	Y
15	heart-h	294	14	5	Y	Y
16	heart-statlog	270	14	2	N	Y
17	hepatitis	155	20	2	Y	Y
18	hypothyroid	3772	30	4	Y	Y
19	ionosphere	351	35	2	N	Y
20	iris	150	5	3	N	Y
21	kr-vs-kp	3196	37	2	N	N
22	labor	57	17	2	Y	Y
23	letter	20000	17	26	N	Y
24	lymph	148	19	4	N	Y
25	mushroom	8124	23	2	Y	N
26	primary-tumor	339	18	21	Y	N
27	segment	2310	20	7	N	Y
28	sick	3772	30	2	Y	Y
29	sonar	208	61	2	N	Y
30	soybean	683	36	19	Y	N
31	splice	3190	62	3	N	N
32	vehicle	846	19	4	N	Y
33	vote	435	17	2	Y	N
34	vowel	990	14	11	N	Y
35	waveform-5000	5000	41	3	N	Y
36	zoo	101	18	7	N	Y

AUC [11]–[13]). The classification accuracy and AUC of each tree on each data set is obtained via 10 runs of 10-fold cross-validation. Runs with the various tree algorithms are carried out on the same training sets and evaluated on the same test sets. In particular, the cross-validation folds are the same for all the experiments on each data set. Finally, we conducted a two-tailed t-test with 95% confidence level [14] to compare the Laplace estimate with the other estimates.

Table II - Table III respectively shows the classification accuracy and AUC scores of each tree on each data set, and the symbols v and * in the tables respectively denote statistically significant upgradation or degradation over the Laplace estimate with a 95% confidence level. Besides, The averages and $w/t/l$ values are summarized at the bottom of the tables. Each entry $w/t/l$ in the table means that the other estimates win on w data sets, tie on t data sets, and lose on l data sets, compared to the Laplace estimate. Now, we summarize some highlights briefly as follows:

1) There is no significant difference between the Laplace estimate and the m-estimate in terms of accuracy and AUC. The $w/t/l$ values respectively is 0/36/0 and 1/34/1. This fact proves that the classification and ranking performance of decision trees with different class probability estimation methods are no significant difference if only the class variable is used.
2) In terms of accuracy and AUC, the similarity-weighted estimate and the naive Bayes-based estimate significantly outperform the Laplace estimate. The $w/t/l$ values respectively is 15/15/6, 16/18/2, 19/17/0, and 17/19/0. This fact proves that taking some attribute variables into the class probability estimation, instead of only using the class variable, can scale up the classification and ranking performance of decision trees.
3) In terms of accuracy and AUC, a bagging of the Laplace estimate significantly outperform the Laplace estimate. The $w/t/l$ values respectively is 22/14/0 and 23/13/0. This fact proves that applying bagging etc. ensemble learning methods and averaging the class-membership probabilities from multiple decision trees, instead of estimating the class-membership probabilities from a single tree, can also scale up the classification and ranking performance of decision trees.

Besides, in our another experiments, we compare the classification and ranking performance of C4.4 [2] with boosted C4.4 [8], [15] and Random Forest [16], and surprisedly found that Random Forest almost ties C4.4 in terms of ranking (9 wins and 6 losses). Due to the

TABLE II.
EXPERIMENTAL RESULTS ON CLASSIFICATION ACCURACY AND STANDARD DEVIATION.

Dataset	LE	ME	SWE		NBE		LE-Bagging	
anneal	99.57±0.67	99.57±0.67	98.9±1.28		99.4±0.73		99.12±0.8	
anneal.ORIG	90.88±2.59	90.88±2.59	90.43±2.56		91.06±2.55		92.51±2.18	v
audiology	83.28±7.56	83.28±7.56	73.86±8.75	*	79.04±8.12	*	82.31±7.54	
autos	80.66±8.12	80.66±8.12	71.74±10.21	*	79.59±8.35		83.23±7.95	
balance-scale	63.62±3.65	63.62±3.65	67.17±4.54	v	71.23±4.43	v	76.37±4.29	v
breast-cancer	65.39±7.76	65.39±7.76	72.95±6.38	v	71.41±6.98	v	68.77±7.76	
breast-w	91.7±3.09	91.7±3.09	93.99±2.96	v	94.16±2.97	v	95.44±2.75	v
colic	78.42±6.21	78.42±6.21	83.23±5.62	v	83.23±6.18	v	83.17±5.88	v
colic.ORIG	75.62±6.44	75.62±6.44	79.59±6.49	v	80.65±6.37	v	79.59±5.58	
credit-a	77.8±4.11	77.8±4.11	83.86±3.74	v	83.57±3.94	v	83.81±4.29	v
credit-g	67.75±3.91	67.75±3.91	70.86±3.9	v	70.58±3.77		72.12±3.81	v
diabetes	68.57±3.5	68.57±3.5	72.93±4.99	v	71.98±5.09	v	72.6±4.71	v
glass	56.38±9.15	56.38±9.15	58.04±9.11		58.4±9.29		59.34±9.16	
heart-c	73.36±8.67	73.36±8.67	75.65±7.73		75.06±7.19		78.45±6.85	v
heart-h	74.31±6.93	74.31±6.93	76.23±7.45		75.56±7.33		79±7.01	v
heart-statlog	74.11±8.03	74.11±8.03	73.56±7.63		76.3±6.64		77.52±7.22	
hepatitis	77.1±10.63	77.1±10.63	82.6±8.64		80.54±10.24		81.75±9.42	
hypothyroid	91.51±0.83	91.51±0.83	92.73±0.73	v	92.81±0.74	v	92.27±0.78	v
ionosphere	83.85±5.57	83.85±5.57	90.25±4.9	v	87.49±5.25	v	90.22±4.66	v
iris	90±7	90±7	96±4.64	v	95.87±4.72	v	94.87±5.51	v
kr-vs-kp	99.62±0.33	99.62±0.33	99.06±0.44	*	99.29±0.45	*	99.53±0.41	
labor	82.23±15.51	82.23±15.51	89.77±11.68		89.4±10.89		87.97±12.72	
letter	79.98±0.88	79.98±0.88	78.38±0.86	*	81.95±0.77	v	84.09±0.83	v
lymph	71.22±10.64	71.22±10.64	76.04±9.46		80.1±9.17	v	80.77±9.59	v
mushroom	100±0	100±0	100±0		100±0		100±0	
primary-tumor	36.11±7.04	36.11±7.04	41.8±6.59	v	43.81±6.17	v	41.12±6.32	v
segment	92.74±1.63	92.74±1.63	92.16±1.78		93.03±1.71		94.44±1.57	v
sick	98.02±0.75	98.02±0.75	98.1±0.69		98.1±0.67		97.93±0.72	
sonar	66.55±9.64	66.55±9.64	71.48±9.7		71.28±10.9		75.03±8.75	v
soybean	92.62±3.01	92.62±3.01	92.02±2.77		94.09±2.58		93±2.87	
splice	90.03±1.63	90.03±1.63	93.35±1.27	v	93.12±1.25	v	94.65±1.16	v
vehicle	67.58±4.24	67.58±4.24	69±3.88		69.87±3.75		71.73±3.73	v
vote	93.05±3.26	93.05±3.26	95.61±2.75	v	94.18±3.46		95.52±2.94	v
vowel	76.91±4.19	76.91±4.19	68.72±4.47	*	82.98±3.91	v	81.71±3.92	v
waveform-5000	64.91±2.02	64.91±2.02	71.8±1.92	v	71.88±1.93	v	75.19±1.83	v
zoo	96.85±5.51	96.85±5.51	87.16±6.54	*	94.97±6.36		94.79±6.57	
Mean	79.79±5.13	79.79±5.13	81.36±4.918		82.67±4.86		83.61±4.78	
w/t/l	-	0/36/0	15/15/6		16/18/2		22/14/0	

space limit, we don't provide the detailed experimental results here. So, how to improve the ranking performance of Random Forest is our main work in the future.

IV. CONCLUSIONS

A critical problem in decision tree learning is the class probability estimation problem at each leaf node of the tree. In this paper, we provide an empirical study on the classification and ranking performance of the resulting decision tree using different class probability estimation methods.

From our experiments, we can draw conclusions: 1) The classification and ranking performance of decision trees with different class probability estimation methods are no significant difference if only the class variable is used. 2) Taking some attribute variables into the class probability estimation and averaging the class-membership probabilities from multiple decision trees can scale up the classification and ranking performance of the built decision trees.

ACKNOWLEDGEMENTS

We thank anonymous reviewers for their valuable comments and suggestions. This research is supported by the National Natural Science Foundation of China under grant no. 60905033, the Natural Science Foundation of Hubei Province under grant no. 2009CDB139, and the Special Fund for Basic Scientific Research of Central Colleges, China University of Geosciences (Wuhan) under grant no. CUG090109.

REFERENCES

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

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

[3] W. D. Jiang, L. and Z. Cai, "Scaling up the accuracy of bayesian network classifiers by m-estimate," ser. Proceedings of the 3rd International Conference on Intelligent Computing. Springer, 2007, pp. 475–484.

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

[5] 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.

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

TABLE III.
EXPERIMENTAL RESULTS ON AUC AND STANDARD DEVIATION.

Dataset	LE	ME		SWE		NBE		LE-Bagging	
anneal	94.33±2.36	94.34±2.36		96.14±1.19	v	96.18±1.4	v	94.93±2.26	
anneal.ORIG	94.17±1.94	94.17±1.94		95.26±1.94	v	95.03±2.1		94.92±1.83	v
audiology	70.33±0.71	70.37±0.72		71.16±0.7	v	71.01±0.72	v	71.09±0.72	v
autos	91.82±3.42	91.81±3.46		93.66±2.59		93.56±3.94		94.58±2.56	v
balance-scale	64.11±6.69	63.93±6.93		59.4±8.49		63.16±7.8		71.26±7.58	v
breast-cancer	59.03±10	59.03±10		62.09±11.89		61.83±11.64		62.75±10.23	
breast-w	98.04±1.31	98.04±1.31		98.54±1.12		98.74±1.06	v	98.76±1.08	v
colic	81.69±8.23	81.69±8.23		84.6±6.9		86.93±6.79	v	86.98±6.95	v
colic.ORIG	81.72±6.65	81.72±6.65		83.5±7.16		84.11±7.06		84.77±5.76	
credit-a	86.71±3.82	86.71±3.82		89.2±3.79	v	89.9±3.69	v	89.55±3.73	v
credit-g	68.37±4.75	68.37±4.75		70.55±4.39		71.61±4.35	v	73.32±4.98	v
diabetes	74.31±4.91	74.31±4.91		77.28±5.54	v	77.45±5.44	v	78.27±5.73	v
glass	78.49±6.55	78.98±6.23		85.73±4.17	v	82.14±5.68		82.5±5.7	v
heart-c	83.04±0.84	83.04±0.84		83.3±0.74		83.32±0.72		83.6±0.64	v
heart-h	83.2±0.75	83.2±0.75		83.37±0.74		83.42±0.69		83.63±0.68	v
heart-statlog	79.48±9.65	79.48±9.65		81.49±8.88		82.33±8		84.8±7.81	v
hepatitis	75.82±14.33	75.82±14.33		77.44±15.55		80.54±13.82		81.62±14.36	
hypothyroid	83.19±7.68	83.4±7.99		84.33±7.97		82.74±8.23		83.03±7.35	
ionosphere	91.62±5.39	91.62±5.39		93.4±4.25		94.17±3.93		95.22±3.82	v
iris	97.22±2.77	97.27±2.75		98.97±1.68	v	99.01±1.56		98.59±2.31	
kr-vs-kp	99.96±0.06	99.96±0.06		99.95±0.06		99.88±0.18		99.97±0.05	
labor	84.83±18.23	84.83±18.23		92.67±17.09		94.75±14.73		87.67±19.81	
letter	96.58±0.29	96.5±0.3	*	97.64±0.23	v	98.12±0.18	v	98.22±0.21	v
lymph	86.15±5.01	86±4.94		89.67±2.26	v	89.4±2.36	v	89.15±2.33	
mushroom	100±0	100±0		100±0	v	100±0	v	100±0	
primary-tumor	74.75±2.02	74.9±2.06		78.46±1.89	v	77.82±2.16	v	77.72±2.05	v
segment	99.04±0.34	99.04±0.34		99.23±0.31	v	99.36±0.29	v	99.31±0.28	v
sick	99.14±0.53	99.15±0.53		99.03±0.51		98.71±1.1		99.2±0.5	
sonar	76.22±8.77	76.22±8.77		75.52±10.76		80.74±10.71		83.03±9.68	v
soybean	91.38±1.59	91.42±1.58		99.28±0.85	v	99.59±0.53	v	92.7±1.55	v
splice	97.93±0.7	97.93±0.7		98.45±0.61	v	98.25±0.74		98.79±0.55	v
vehicle	84.33±2.92	84.2±2.94		86.54±2.68	v	87.2±2.54	v	88.61±2.42	v
vote	97.27±2.68	97.27±2.68		98.36±1.79	v	98.42±1.55		98.26±1.97	
vowel	91.93±2.24	92.35±2.22	v	94.68±1.55	v	97±1.3	v	95.98±1.57	v
waveform-5000	80.64±1.45	80.68±1.46		87.52±1.22	v	86.96±1.25	v	89.81±1.3	v
zoo	88.33±2.81	88.33±2.81		89.42±2.38	v	89.45±2.38	v	88.36±2.76	
Mean	85.70±4.23	85.72±4.24		87.66±4.00		88.13±3.91		88.36±3.98	
w/t/l	-	1/34/1		19/17/0		17/19/0		23/13/0	

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

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

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

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

[11] A. P. Bradley, "The use of the area under the roc curve in the evaluation of machine learning algorithms," *Pattern Recognition*, vol. 30, pp. 1145–1159, 1997.

[12] 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.

[13] H. J. Ling, C. X. and H. Zhang, "Auc: a statistically consistent and more discriminating measure than accuracy," ser. Proceedings of the International Joint Conference on Artificial Intelligence. Morgan Kaufmann, 2003, pp. 329–341.

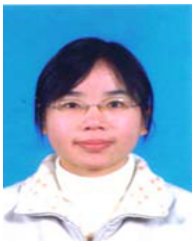
[14] C. Nadeau and Y. Bengio, "Inference for the generalization error," *Machine Learning*, vol. 52, pp. 239–281, 2003.

[15] R. E. S. Y. Freund, "Experiments with a new boosting algorithm," ser. Proceedings of the 13th International Conference on Machine Learning. Morgan Kaufmann, 1996, pp. 148–156.

[16] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

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.