Ranking the Rules and Instances of Decision Trees *

Yuh-Jye Lee and Yi-Ren Yeh

Dept. of Computer Science & Information Engineering,
National Taiwan University of Science & Technology, Taipei, Taiwan
{yuh-jye, M9315027}@mail.ntust.edu.tw

Abstract. Traditionally, decision trees rank instances by using the local probability estimations for each leaf node. The instances in the same leaf node will be estimated with equal probabilities. In this paper, we propose a hierarchical ranking strategy by combining decision trees and leaf weighted Naïve Bayes to improve the local probability estimation for a leaf node. We consider the importance of the rules, and then rank the instances fit in with the rules. Because the probability estimations based on Naïve Bayes might be poor, we investigate some different techniques which were proposed to modify Naïve Bayes as well. Experiments show that our proposed method has significantly better performance than that of other methods according to paired t-test. All results are evaluated by using AUC (Area under ROC Curve) instead of classification accuracy.

1 Introduction

In many machine learning problems, the goal is not only correctly classifying the instances. Ranking instances based on the class probabilities is more desirable for practical applications. For example, we would like to rank the resulting pages returned by a search engine to match the user's preference. In other words, the requirements of some applications and problems are not only the boolean response of "positive or negative", but also the ranking according to the probability of being positive. Most learning models provide the information of the probability estimation which might be used to rank instances. How to deal with those information from learning models is an interesting research issue [17, 7, 13, 15]. We will focus on ranking the rules and instances generated by decision trees model. We will use the term "local" to indicate that the estimation is based on a particular leaf node information. Thus the local probability estimation of $P(c|x)$ is given by $\frac{k}{n}$, where k is the number of training instances of class c in the leaf node and n is the total number of training instances in the leaf node. However,

* This work was supported in part by the National Science Council under the Grants NSC-93-2213-E-001-013, NSC-93-2422-H-001-0001, and NSC-93-2752-E-002-005-PAE, and by the Taiwan Information Security Center (TWISC), National Science Council under the Grants NSC 94-3114-P-001-001-Y and NSC 94-3114-P-011-001.

decision trees built by C4.5 [19] have been observed to produce poor probability estimations [18], even though they have a good performance in accuracy and an intelligible result. The local probability estimated by this way is rough and unreasonable. There are two main drawbacks of this intuitive approximation. The instances in the same leaf node will be estimated with the same probability. There is no difference between each individual instance in the same leaf node. Moreover, there is no difference between the leaf nodes which have the same proportion of positive instances but have different sizes. A more sophisticated way will be needed for the local probability estimation. In this paper, we describe a hierarchical ranking strategy by combining decision trees and Naïve Bayes. In the first step, we apply Laplace smoothing technique to re-estimate local probability for each leaf node. This will differentiate the leaf nodes which have the same proportion of positive class instances. We will rank the rules based on these local probabilities ordering. This also gives a grouping order for every instance. In the second step, we build a *leaf Naïve Bayes classifier*, the Naïve Bayes classifier generated by the instances in a leaf node, for each leaf node to rank the instances in the leaf node. However, building a Naïve Bayes classifier for a small training set tends to having a more serious *zero frequency* problem [21]. We will use the information provided by the global Naïve Bayes classifier which is generated by the entire training set as the prior knowledge in applying smoothing technique to the leaf Naïve Bayes classifier. Thus, we can rank every instance in a hierarchical way. We test our proposed methods on 7 public available datasets from UCI repository [2]. The experiment results show that our proposed method has significantly better performance than that of other methods according to paired t-test. All results are evaluated by using AUC (Area under ROC Curve) instead of classification accuracy.

The paper is organized as follows. In Section 2, we discuss some related works on improving decision trees ranking. Section 3 describes our hierarchical ranking method in details. We report the experiment results and discuss the details of tuning the parameters of our proposed method in Section 4. We conclude the paper with a brief discussion in Section 5.

2 Related Work

Traditional decision trees algorithms, such as C4.5 and ID3, have been observed to produce poor probability estimations. One of the reasons is that the algorithms aim at building a small and accurate tree which biases against good probability estimations [17]. Many techniques have been proposed to improve the probability estimations of decision trees [7, 16, 22, 12, 1]. Most of them apply the smoothing methods [7, 16, 22] to modify the local probability estimation. They correct the traditional probability estimation by some corrected ratio that shifts the probability toward the prior probability of the class. The most popular smoothing method is Laplace smoothing which estimates the probability $P(c|x)$ by $\frac{k+1}{n+|C|}$ at a leaf node, where k is the number of training instances of class c in the leaf node, n is the total number of training instances in the leaf

node, and $|C|$ is the number of classes. Laplace smoothing uses a uniform class distribution, $\frac{1}{|C|}$, as a prior knowledge to correct the reliability of probability estimation. The more instances a leaf node owns, the less the prior knowledge can affect. This can distinguish those leaf nodes which have the same proportion of positive instances but have different sizes. Although the smoothing methods improve the probability estimation by considering the reliability of the leaf node, the instances in the same leaf node still have the same probability estimations.

Other methods, like [12, 1], suggest that the probability estimations of the instances in the same leaf should be different. In [12], they average probability estimations from all leaves of the tree to produce different probability estimations in the same leaf node. However, this method only uses the attributes on the tree. It might easily produce duplicate points in pure nominal datasets, especially with a high dimensional dataset and a small tree. The method in [1] is similar to our proposed method. They propose a geometric score to distinguish the probability estimations of instances in the leaf. This method is limited to numerical attributes.

3 Hierarchical Ranking of Decision Trees

In order to keep the intelligibility and replace the same probability estimation in the leaf node, we describe a hierarchical ranking strategy by combining decision trees and Naïve Bayes. In first step, we rank the rules, and secondly we rank the instances in the leaf nodes.

3.1 Ranking Rules and Ranking Instances in a Leaf Node

In the first step, we adopt the smoothing techniques to rank the rules produced by decision trees. As described in section 2, many smoothing techniques have been proposed to improve the ranking of leaf nodes. The smoothing techniques will help us to differentiate the leaf nodes which have the same proportion of positive class instances without destroying the tree structure. We note that these smoothing techniques are often applied to unpruned trees in order to produce more different probability estimations. This might reduce the intelligibility of the tree because of the more complex rules. In our proposed method, Laplace smoothing is applied to the pruned trees directly. We then distinguish instances in the same leaf node via an embedded leaf Naïve Bayes.

Distinguishing instances in the same leaf node is an important part of decision trees ranking. In order to achieve this goal, we adopt the following strategies in the second step:

- Embedding a Naïve Bayes classifier in each leaf node,
- Combining leaf Naïve Bayes classifier and global Naïve Bayes in estimating $P(a_i|c)$, where a_i is the ith attribute and c is the given class,
- Using weighted Naïve Bayes classifier to incorporate the information provided by *non-tree* attributes.

We will discuss them in details in the following subsections.

3.2 Leaf Naïve Bayes Classifier

Naïve Bayes usually works well when tested on actual datasets, particularly combined with some of the attribute selection procedures [21]. Like decision trees, it also can deal with hybrid data types by the normal-distribution assumption of numerical attributes. For these reasons, we embed Naïve Bayes in each leaf node to rank instances. Since our goal is to distinguish the probability estimations of instances in a leaf node, the information of instances in the leaf node should be more important than that of the entire training set. Thus, we only use the instances in the leaf node to generate the leaf Naïve Bayes mainly and the information given by the global Naïve Bayes will be treated as prior knowledge for the leaf Naïve Bayes.

3.3 Estimating $P(a_i|c)$

Leaf Naïve Bayes can describe a good local distribution of $P(a_i|c)$ in the leaf node, but has less reliability due to the smaller size of training instances in a leaf node. In order to keep the local information and raise the reliability, we take the information from global Naïve Bayes into account when estimating $P(a_i|c)$ in leaf Naïve Bayes. For the flexibility, we also harmonize the information from leaf Naïve Bayes and global Naïve Bayes with a varied weight. How to determine the weight will be discussed more in section 4. Since Naïve Bayes adopts different strategies in estimating $P(a_i|c)$ for numerical and nominal attributes, we will describe our combination method for the two type attributes respectively.

In estimating $P(nominal_i|c)$, leaf Naïve Bayes will cause zero frequency more seriously. Traditionally, Naïve Bayes use Laplace smoothing to overcome zero frequency problem. Instead of using uniform prior knowledge in Laplace smoothing, we use m-estimate smoothing technique [7] to carry the prior knowledge from the entire training set as follows:

$$P(nominal_i|c) = \frac{k_i + w \cdot t_i}{N + w \cdot M} \tag{1}$$

where $\frac{k_i}{N}$ and $\frac{t_i}{M}$ are the estimations of $P(nominal_i|c)$ in the leaf Naïve Bayes and global Naïve Bayes respectively. The w is a nonnegative weight parameter for controlling the importance of the prior knowledge given by global Naïve Bayes. It can be varied for each leaf node.

The smoothing technique is only suitable for nominal attributes. It means that we need a new strategy to combine the information of the leaf Naïve Bayes and global Naïve Bayes in estimating the probability density measure for the numerical attributes. The probability density measure of ith attribute is denoted by $P(numerical_i|c)$ if the ith attribute is numerical type. We abused the notation a little bit here. Generally, Naïve Bayes conducts numerical attributes with normal-distribution assumption, which can be determined by the sample mean and sample standard deviation of the numerical attribute. The local probability is replaced by a local probability density measure. Inspired by the idea in dealing with nominal attributes, we combine the sample mean and sample standard

deviation of $P(numerical_i|c)$ in the leaf Naïve Bayes , denoted by Mean_{local} and Std_{local}, and that in the global Naïve Bayes, denoted by Mean_{global} and Std_{global}, with convex combination. That is:

$$\begin{aligned} \text{Mean}_i &= (1-w)\cdot \text{Mean}_{local} + w \cdot \text{Mean}_{global} \\ \text{Std}_i &= (1-w)\cdot \text{Std}_{local} + w \cdot \text{Std}_{global} \end{aligned} \tag{2}$$

where w is the varied weight of the prior knowledge similar to the case of nominal attribute. Finally, the normal distribution of the ith attribute given the class c in the leaf node is determined by Mean_i and Std_i.

3.4 Using Weighted Naïve Bayes

Attributes independent and equally important are the assumptions of Naïve Bayes, but these assumptions will not always hold in real data especially in high dimensional data. In fact, many research focus on breaking these assumptions to improve the performance of Naïve Bayes, like Bayesian network and weighted Naïve Bayes. Bayesian networks focus on breaking the attribute independence assumption, but it still supposes attributes are equally important. On the other hand, weighted Naïve Bayes focuses on breaking attributes equally important and keep the independent assumption. Here, we will describe how weighted Naïve Bayes can break the equally important assumption. The original Naïve Bayes classifier is defined as follows:

$$\text{NB}(x) = \arg\max_c P(c)\prod_{i=1}^{n} P(a_i|c),\ x = (a_1, a_2, \cdots, a_n),\ c : class \tag{3}$$

If we increase the degree of $P(a_i|c)$ in Naïve Bayes classifier, that will enlarge the influence of the attribute a_i(see Table 1). Thus, we can extend Naïve Bayes to weighted Naïve Bayes as follows [10]:

$$\text{WNB}(x) = \arg\max_c P(c)\prod_{i=1}^{n} P(a_i|c)^{w_i},\ x = (a_1, a_2, \cdots, a_n),\ c : class \tag{4}$$

In the weighted Naïve Bayes classifier, we break the equally important assumption by using different weights.

Decision trees are usually built by the subset of all attributes. Intuitively, only using the attributes on the tree in leaf Naïve Bayes should be more representative. In fact, removing redundant attributes will be good for the performance of Naïve Bayes. In other words, only using the attributes on the tree is similar to the attribute selection. However, only using the attributes on the tree will easily produce many duplicate points in pure nominal datasets, especially with a high dimensional dataset and a small tree. We have mentioned this phenomenon in section 2. In order to overcome the duplicate points problem, we use all attributes to build up the leaf Naïve Bayes but give tree attributes more weight. The attribute weighting is closely related to attribute selection. Thus, we also can emphasize the importance of tree attributes by using attribute weighting.

	Degree 1	Degree 2	Degree 3
$P(a_i\|c_1)$	0.5	0.25	0.125
$P(a_i\|c_2)$	0.1	0.01	0.001
difference	5 times	25 times	125 times

Table 1. The effect of different degrees of $P(a_i|c)$

4 Numerical Results and Comparisons

Using accuracy in evaluating models will completely ignore probability estimations produced by learning models. As the purpose is probability estimations or ranking, accuracy is not sufficient in measuring and comparing learning models. The area under the ROC (Receiver Operating Characteristics) curve, or simply AUC, has been shown as an measurement for the quality of ranking [3, 11, 12]. The ROC curve was first used in signal detection theory to represent the tradeoff between the hit rates and false alarm rates [6, 9]. It has been extensively studied and applied in medical diagnosis since 1970s [14, 20]. The AUC can be expressed in a simpler form: if the sample contains m positives and n negative examples, we can denote AUC simply by the following Wilcoxon-Mann-Whitney statistic [4]:

$$AUC = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} I_{P(x_i)>P(x_j)} + \frac{1}{2} I_{P(x_i)=P(x_j)}}{mn} \tag{5}$$

where $P(x)$ is the probability estimation from learning models and I_π is defined to 1 if the predicate π holds and 0 otherwise.

In our experiment, we evaluate our proposed method and compare with other methods on 7 two-class datasets from the UCI repository. The datasets are described in Table 2. In Lymphography dataset, we throw away the 4 instances to reduce the classes. In numerical results, we repeat 5-fold cross validation on each dataset 5 times and report the mean of them.

4.1 Comparing Our Proposed Method with Other Methods

In this section, we will compare our proposed method with C4.4 [17], C4.5, and C4.5-L. The C4.5 is the most popular decision trees algorithm which incorporates with some pruning strategy for a better classification result. While the C4.4 does not prune decision trees, it might cause the over-fitting problem but will give a better ranking result. Another difference between the C4.5 and the C4.4 is that the C4.4 ranks the instances with Laplace smoothing and C4.5 does not. In our experiment the C4.5-L denotes C4.5 with Laplace smoothing. Table 3 shows the numerical results of them. The number before the slash is the AUC score and the other number is the quantity $\delta = \frac{A_1 - A_2}{1 - A_2}$ [5], where A_1 and A_2 are the

Dataset	Size	Nom. Attributes	Num. Attributes
Tic Tac Toe	958	9	0
House Vote	435	16	0
SPECT	267	22	0
Bupa	345	0	6
Ionosphere	351	0	34
Sonar	208	0	60
Lymphography	142	15	3

Table 2. Descriptions of datasets

AUC scores of $Method_1$ and $Method_2$ respectively. The performance metric δ indicates what percentage of $Method_2$'s missing ROC area $(1 - A_2)$ is covered by $Method_1$. In Table 3, we use C4.5 as the counterpart method ($Method_2$) and compute the δ values of using C4.4, C4.5 and our proposed method as the $Method_1$.

As we mentioned above, we need to determine two parameters, weights of prior knowledge and weights of attributes, in our method. The details of tuning heuristics and procedures will be discussed in Section 4.2. In Table 3, we use the optimal parameters followed by Section 4.2, and we can observe that C4.4 is better than C4.5 and C4.5-L. Therefore, we use a paired t-test on our proposed method and C4.4 with a 95% confidence coefficient. The test result shows our method is significantly better than C4.4 in AUC scores, except the dataset Tic Tac Toe (see Table 4). This shows that our method can significantly improve the ranking of decision trees. In fact, our proposed method is C4.5 with smoothing techniques and leaf weighted Naïve Bayes. In other words, we not only can rank the leaf nodes with smoothing techniques but also have the ability to rank the instances in the leaf nodes. Moreover, we also can show the necessity for ranking instances in the leaf nodes because instances with equal probability estimations in the same leaf node will be treated as random ranking within the leaf node (see Fig. 1).

4.2 Parameters Tuning

In order to increase the reliability, we take the information from the global Naïve Bayes into account when estimating $P(a_i|c)$ in a leaf Naïve Bayes. From the idea, there is a point of view to regard for the weight of the global Naïve Bayes. If there are enough instances in the leaf nodes, the weight of prior knowledge should be less. Therefore, we will take a varied weight for each leaf node when adjusting $P(a_i|c)$ via the m-estimate smoothing technique and convex combination. We

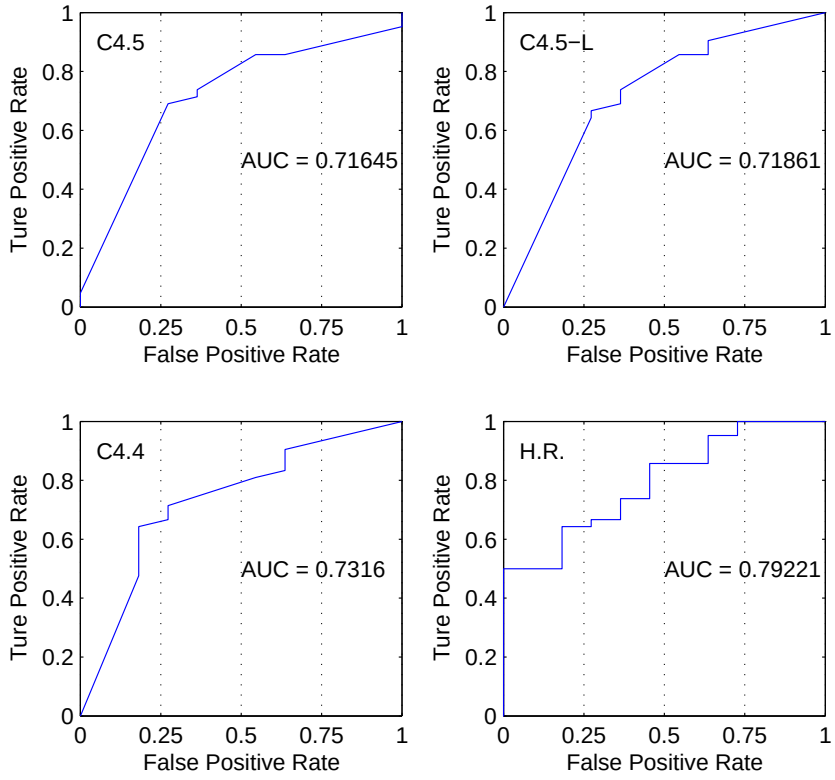

Fig. 1. ROC curves of C4.5 (left top), C4.5-L (right top), C4.4 (left bottom), and our proposed method (right bottom) of a particular fold of SPECT dataset.

define our varied weight to achieve our idea as follows:

$$w_i = 1 - \alpha_i, \text{ where } \alpha_i = \frac{\text{number of instances in the leaf node } i}{\text{number of the entire training instances}} \tag{6}$$

Table 5 shows the results. In the second and third columns, we fixed $w_i = 0$ and $w_i = 1$ for all leaf node i. They represent only using the leaf Naïve Bayes and the global Naïve Bayes respectively. Obviously, the varied $w_i = 1 - \alpha_i$ is more appropriate of them. The result also shows that combining the leaf Naïve Bayes and global Naïve Bayes is good for ranking instances in the leaf node.

As we mentioned in Section 3, removing redundant attributes will be good for the performance of Naïve Bayes. Unfortunately, removing redundant attributes may easily produce duplicate instances in pure nominal datasets. In order to

Dataset	C4.5	C4.5-L	C4.4	Hierarchical R.
Tic Tac Toe	0.8896	0.9050 / 13.9%	**0.9281 / 34.8%**	0.9111 / 19.5%
House Vote	0.9603	0.9706 / 25.9%	0.9746 / 36.0%	**0.9885 / 71.0%**
SPECT	0.7769	0.7978 / 9.4%	0.7782 / 0.5%	**0.8433 / 29.8%**
Bupa	0.6743	0.6884 / 4.3%	0.6942 / 6.1%	**0.7045 / 9.3%**
Ionosphere	0.8922	0.9326 / 37.5%	0.9314 / 36.3%	**0.9526 / 51.6%**
Sonar	0.7322	0.7836 / 19.2%	0.7836 / 19.2%	**0.8137 / 30.4%**
Lymphography	0.8263	0.8471 / 12.0%	0.8554 / 16.8%	**0.8877 / 35.4%**

Table 3. Numerical results of C4.5, C4.5-L, C4.4, and our proposed hierarchical ranking method on seven public available datasets in UCI repository

Dataset	p-value
Tic Tac Toe	0.9996
House Vote	0.0108
SPECT	0.0434
Bupa	0.0287
Ionosphere	0.0125
Sonar	0.0238
Lymphography	0.0193

Table 4. The p values of the paired t-test on our proposed method and C4.4

overcome this problem, we use all attributes but decrease the weight of non-tree attributes in leaf weighted Naïve Bayes. Table 6 shows the results of different weights on non-tree attributes. We fix the weight of tree attributes and vary the weight, w, of non-tree attributes to 0, 0.01, 0.1, 0.5, and 1. $w = 0$ and $w = 1$ means only using tree attributes and all attributes with equal weights respectively. $w = 0.5$ had the best performance in our experiment. It shows that our strategy of weighting is good for the performance. Note that we have better improvement in two pure nominal datasets, House vote and SPECT, by comparing with $w = 0$. In our observation, only using tree attributes will result in producing some equal probability estimations in the leaf node because of dimension reduction. It means that using weighted Naïve Bayes will not only raise the ability of Naïve Bayes, but also solve the problem of dimension reduction.

Dataset	$w = 0$	$w = 1$	$w = 1 - \alpha$
Tic Tac Toe	0.9110	0.9079	0.9111
House Vote	0.9792	0.9885	0.9885
SPECT	0.8286	0.8437	0.8433
Bupa	0.7020	0.7042	0.7045
Ionosphere	0.9441	0.9489	0.9526
Sonar	0.8046	0.8135	0.8137
Lymphography	0.8770	0.8902	0.8877

Table 5. The effect of different prior knowledge weights in estimating $P(a_i|c)$, where α is defined by (6). In this table, we use the same weight 0.5 of non-tree attributes for all.

Dataset	$w = 0$	$w = 0.01$	$w = 0.1$	$w = 0.5$	$w = 1$
House Vote	0.9790	0.9883	0.9881	0.9885	0.9880
SPECT	0.8310	0.8336	0.8416	0.8433	0.8403
Ionosphere	0.9471	0.9497	0.9507	0.9526	0.9518
Sonar	0.8089	0.8096	0.8113	0.8137	0.8140
Lymphography	0.8846	0.8882	0.8882	0.8877	0.8769

Table 6. Different weights of non-tree attributes. In this table, we use the same varied weights of prior knowledge for all.

5 Conclusion and Future Work

In this paper, we present a hierarchical ranking method for decision trees. We rank the rules as well as the instances fit in with the rules. This method combines decision trees and Naïve Bayes via embedding leaf weighted Naïve Bayes in each leaf node. The hierarchical ranking strategy will retain the intelligibility of decision trees. Another important feature of our method is that it can deal with hybrid datasets as well. Experiment results show that our proposed method improves the AUC score over other methods. It means that ranking instances in the leaf node works in improving the ranking performance.

Our method needs to set two parameters, weights of prior knowledge and attributes. For the weight of attributes, we simply assign two different weights for tree attributes and non-tree attributes in our experiments. In fact, how to determine the weight of weighted Naïve Bayes is an interesting issue [8]. In our future work, we will study how to determine different weights for each attribute and take the depth of the split attribute in decision trees into account.

References

1. I. Alvarez and S. Bernard. Ranking cases with decision trees: a geometric method that preserves intelligibility. In *Proceedings of 18th Internationa Conference on Artificial Intelligence(IJCAI-2005)*, 2005.
2. C. Blake and C. Merz. Uci repository of machine learning databases. *niversity of California (http://www.ics.uci.edu/∼mlearn/MLRepository.html)*, 1998.
3. A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30:1145–1159, 1997.
4. C. Cortes and M. Mohri. Auc optimization vs. error rate minimization. In *Advances in Neural Information Processing Systems(NIPS 2003)*, 2004.
5. D. Dash and G. F. Cooper. Model averaging for prediction with discrete bayesian networks. *Journal of Machine Learning Research*, 5:1177–1203, 2004.
6. J. Egan. Signal detection theory and roc analysis. *New York:Academic Pres*, 1975.
7. C. Ferri, P. Flach, and J. Hernández-Orallo. Improving the auc of probabilistic estimators trees. In *Proceedings of 14th European Conference on Machine Learning(ECML'2003)*, pages 121–132, 2003.
8. T. Gärtner and P. A. Flach. Wbcsvm: Weighted bayesian classification based on support vector machines. In *Proceedings of the 18th International Conference on Machine Learning*, pages 207–209, 2001.
9. D. Green and J. Swets. Signal detection theory and psychophysics. *New York:Wiley*, 1966.
10. S. Sheng H. Zhang. Learning weighted naive bayes with accurate ranking. In *Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM'04)*, pages 567–570, 2004.
11. D. J. Hand and R. J. Till. A simple generalisation of the area under the roc curve for multiple class classification problems. *Machine Learning*, 45:171–186, 2001.
12. C. X. Ling J., Hunag, and H. Zhang. Auc: a astatistically consistent and more discriminating measure than accuracy. In *Proceedings of 18th Internationa Conference on Artificial Intelligence(IJCAI-2003)*, 2003.
13. C. X. Ling and J. Yan. Decision tree with better ranking. In *Proceedings of 2003 International Conference on Machine Learning (ICML'2003)*, pages 480–487, 2003.
14. C. Metz. Basic principles of roc analysis. *Seminars in NuclearMedcine*, 8:283–298, 1978.
15. J. Platt. Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. *in Advances in Large Margin Classifiers, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press*, 2000.
16. F. Provost and P. Domingos. Well-trained pets: Improving probability estimation trees. *Technical Report CDER #00-04-IS*, 2000.
17. F. Provost and P. Domingos. Tree induction for probability-based rankings. *Machine Learning*, 52:199–215, 2003.
18. F. Provost, T. Fawcett, and R. Kohavi. The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the fifteenth international conference on machine learning*, pages 445–453, 1998.
19. J. R. Quinlan. Improved use of continuous attributes in C4.5. *Journal of Artificial Intelligence Research*, 4:77–90, 1996.
20. J. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293, 1988.

21. I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann, San Francisco, CA, 2000. http://www.cs.waikato.ac.nz/∼ml/index.html.
22. B. Zadrozny and C. Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *Proceedings of 18th International Conference on Machine Learning*, pages 609–616, 2001.