AUC: a Better Measure than Accuracy in Comparing Learning Algorithms

Charles X. Ling[1], Jin Huang[1], and Harry Zhang[2]

[1] Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7
E-mail: {ling, jhuang}@csd.uwo.ca
[2] Faculty of Computer Science
University of New Brunswick
Fredericton, NB, Canada E3B 5A3
E-mail: hzhang@unb.ca

Abstract. Predictive accuracy has been widely used as the main criterion for comparing the predictive ability of classification systems (such as C4.5, neural networks, and Naive Bayes). Most of these classifiers also produce probability estimations of the classification, but they are completely ignored in the accuracy measure. This is often taken for granted because both training and testing sets only provide class labels. In this paper we establish rigourously that, even in this setting, the area under the ROC (Receiver Operating Characteristics) curve, or simply AUC, provides a better measure than accuracy. Our result is quite significant for three reasons. First, we establish, for the first time, rigourous criteria for comparing evaluation measures for learning algorithms. Second, it suggests that AUC should replace accuracy when measuring and comparing classification systems. Third, our result also prompts us to re-evaluate many well-established conclusions based on accuracy in machine learning. For example, it is well accepted in the machine learning community that, in terms of predictive accuracy, Naive Bayes and decision trees are very similar. Using AUC, however, we show experimentally that Naive Bayes is significantly better than the decision-tree learning algorithms.

1 Introduction

In classification, the goal of a learning algorithm is to build a classifier from a set of training examples with class labels. The predictive ability of the algorithm is typically measured by its predictive accuracy (or error rate, which is 1 minus the accuracy) on the testing examples. However, most classifiers (including C4.5 and Naive Bayes) can also produce probability estimations or “confidence” of the class prediction. Unfortunately, this information is completely ignored in accuracy. That is, the accuracy measure does not consider the probability (be it 0.55 or 0.99) of the prediction; as long as the class with the largest probability estimation is the same as the target, it is regarded as correct. This is often taken for granted since the true probability is unknown for the testing examples anyway.

In many applications, however, accuracy is not enough. For example, in direct marketing, we often need to promote the top X% of customers during gradual roll-out, or we often deploy different promotion strategies to customers with different likelihood of buying some products. To accomplish these tasks, we need more than a mere classification of buyers and non-buyers. We need (at least) a ranking of customers in terms of their likelihoods of buying. Thus, a ranking is much more desirable than just a classification [1], and it can be easily obtained since most classifiers do produce probability estimations that can be used for ranking (testing) examples.

If we want to achieve a more accurate ranking from a classifier, one might naturally expect that we must need the true ranking in the training examples [2]. In most scenarios, however, that is not possible. Instead, what we are given is a dataset of examples with class labels only. Thus, given only classification labels in training and testing sets, are there better methods than accuracy to evaluate classifiers that also produce rankings?

AUC (area under the curve) of the ROC (Receiver Operating Characteristics) curve has been recently used as an alternative measure for machine learning algorithms [3–5]. The ROC curve compares the classifiers' performance across the entire range of class distributions and error costs. See [3, 5] for more details. Figure 1 shows a plot of four ROC curves, each representing one of the four classifiers, A through D. A ROC curve X is said to *dominate* another ROC curve Y if X is always above and to the left of Y. This means that the classifier of X always has a lower expected cost than that of Y, over all possible error costs and class distributions. In this example, A and B dominate D. However, often

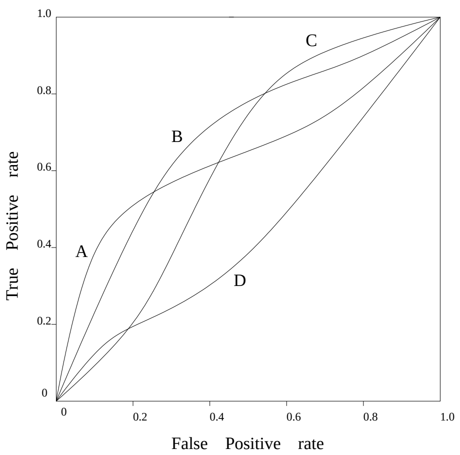

Fig. 1. An example of four ROC curves

there is no clear dominating relation between two ROC curves. For example, curves A and B are not dominating each other in the whole range. In those situations, the area under the ROC curve, or simply AUC, is a good "summary" for comparing the two ROC curves.

Hand and Till [6] present a simple approach to calculating the AUC of a classifier G below.

$$\hat{A} = \frac{S_0 - n_0(n_0+1)/2}{n_0 n_1}, \tag{1}$$

where n_0 and n_1 are the numbers of positive and negative examples respectively, and $S_0 = \sum r_i$, where r_i is the rank of i_{th} positive example in the ranked list. Table 1 shows an example of how to calculate AUC from a ranked list with 5 positive examples and 5 negative examples. The AUC of the ranked list in Table 1 is $\frac{(5+7+8+9+10)-5\times 6/2}{5\times 5}$, which is 24/25. It is clear that AUC obtained by Equation 1 is a measure for the quality of ranking, as the more positive examples are ranked higher (to the right of the list), the larger the term $\sum r_i$. AUC is also shown to be equivalent to the Wilcoxon statistic rank test [7].

Table 1. An example for calculating AUC with r_i

	−	−	−	−	+	−	+	+	+	+
i					1		2	3	4	5
r_i					5		7	8	9	10

Bradley [7] has compared popular machine learning algorithms using AUC, and found that AUC exhibits several desirable properties compared to accuracy. For example, AUC has increased sensitivity in Analysis of Variance (ANOVA) tests, is independent to the decision threshold, and is invariant to *a priori* class probability distributions [7]. However, no formal arguments or criteria have been established. How can we compare two evaluation measures for learning algorithms? How can we establish that one measure is better than another? In this paper, we give formal definitions on the consistency and discriminancy for comparing two measures. We show, empirically and formally, that AUC is indeed a statistically consistent and more discriminating measure than accuracy.

Our work is quite significant for several reasons. First, we establish rigourously, for the first time, that even given only labelled examples, AUC is a better measure (defined in Section 2.1) than accuracy. Second, our result suggests that AUC should replace accuracy in comparing learning algorithms in the future. Third, our results prompt and allow us to re-evaluate well-established results in machine learning. For example, extensive experiments have been conducted and published on comparing, in terms of accuracy, decision tree classifiers to Naive Bayes classifiers. A well-established and accepted conclusion in the machine learning community is that those learning algorithms are very similar as measured by accuracy [8–10]. Since we have established that AUC is a better measure, are those learning algorithms still very similar as measured by AUC? We perform extensive experimental comparisons to compare decision trees and Naive Bayes, and we conclude that Naive Bayes is significantly better than decision tree algorithms in AUC. These kinds of new conclusions are very useful to the machine learning community, as well as to machine learning applications (e.g., data mining). Fourth, as a new measure (such as AUC) has been found and proved to be better than a previous measure (such as accuracy), we can re-design most learning algorithms to optimize the new measure. This would produce classifiers that not only perform well in the new measure, but also in the previous measure, compared to the classifiers that optimize the previous measure, as shown in [1]. This would further improve the performance of our learning algorithms.

2 Comparing Evaluation Measures for Learning Algorithms

Intuitively, we can see why AUC is a better measure than accuracy from the following example. Let us consider two classifiers, Classifier 1 and Classifier 2, both producing probability estimates for a set of 10 testing examples. Assume that both classifiers classify 5 of the 10 examples as positive, and the other 5 as negative. If we rank the testing examples according to increasing probability of being + (positive), we get the two ranked lists as in Table 2.

Table 2. An Example in which two classifiers have the same classification accuracy, but different AUC values

Classifier 1	$- - - - +$ | $- + + + +$
Classifier 2	$+ - - - -$ | $+ + + + -$

Clearly, both classifiers produce an error rate of 20% (one false positive and one false negative), and thus the two classifiers are equivalent in terms of error rate. However, intuition tells us that Classifier 1 is better than Classifier 2, since overall positive examples are ranked higher in Classifier 1 than 2. If we calculate AUC according to Equation 1, we obtain that the AUC of Classifier 1 is $\frac{24}{25}$ (as seen in Table 1), and the AUC of Classifier 2 is $\frac{16}{25}$. Clearly, AUC does tell us that Classifier 1 is indeed better than Classifier 2.

Unfortunately, "counter examples" do exist, as shown in Table 3 on two other classifiers: Classifier 3 and Classifier 4. It is easy to obtain that the AUC of Classifier 3 is $\frac{21}{25}$, and the AUC of Classifier 4 is $\frac{16}{25}$. However, the error rate of Classifier 3 is 40%, while the error rate of Classifier 4 is only 20% (again we assume that the threshold for accuracy is set at the middle so that 5 examples are predicted as positive and 5 as negative). Therefore, a larger AUC does not always imply a lower error rate.

Table 3. A counter example in which one classifier has higher AUC but lower classification accuracy

Classifier 3	$- - - + + \mid - - + + +$
Classifier 4	$+ - - - - \mid + + + + -$

Another intuitive argument for why AUC is better than accuracy is that AUC is more discriminating than accuracy since it has more possible values. More specifically, given a dataset with n examples, there is a total of only $n+1$ different classification accuracies ($0/n$, $1/n$, ..., n/n). On the other hand, assuming there are n_0 positive examples and n_1 negative examples ($n_0+n_1 = n$), there are $n_0n_1 + 1$ different AUC values ($0/n_0n_1$, $1/n_0n_1$, ..., n_0n_1/n_0n_1), generally more than $n+1$. However, counter examples also exist in this regard. Table 4 illustrates that classifiers with the same AUC can have different accuracies. Here, we see that Classifier 5 and Classifier 6 have the same AUC($\frac{3}{5}$) but different accuracies (60% and 40% respectively). In general, a measure with more values is not necessarily more discriminating. For example, the weight of a person (having infinitely many possible values) has nothing to do with the number of siblings (having only a small number of integer values) he or she has.

Table 4. A counter example in which two classifiers have same AUC but different classification accuracies

Classifier 5	$- - + + - \mid + + - - +$
Classifier 6	$- - + + + \mid - - + - +$

How do we compare different evaluation measures for learning algorithms? Some general criteria must be established.

2.1 (Strict) Consistency and Discriminancy of Two Measures

Intuitively speaking, when we discuss two different measures f and g on evaluating two learning algorithms A and B, we want at least that f and g be *consistent* with each other. That is, when f stipulates that algorithm A is (strictly) better than B, then g will not say B is better than A. Further, if f is more *discriminating* than g, we would expect to see cases where f can tell the difference between algorithms A and B but g cannot.

This intuitive meaning of consistency and discriminancy can be made precise as the following definitions.

Definition 1 (Consistency) *For two measures f, g on domain Ψ, f, g are (strictly) consistent if there exist no a, $b \in \Psi$, such that $f(a) > f(b)$ and $g(a) < g(b)$.*

Definition 2 (Discriminancy) *For two measures f, g on domain Ψ, f is (strictly) more discriminating than g if there exist a, $b \in \Psi$ such that $f(a) > f(b)$ and $g(a) = g(b)$, and there exist no a, $b \in \Psi$ such that $g(a) > g(b)$ and $f(a) = f(b)$.*

As an example, let us think about numerical marks and letter marks that evaluate university students. A numerical mark gives 100, 99, 98, ..., 1, or 0 to students, while a letter mark gives A, B, C, D, or F to students. Obviously, we regard A > B > C > D > F. Clearly, numerical marks are consistent with letter marks (and vice versa). In addition, numerical marks are more discriminating than letter marks, since two students who receive 91 and 93 respectively receive different numerical marks but the same letter mark, but it is not possible to have students with different letter marks (such as A and B) but with the same numerical marks. This ideal example of a measure f (numerical marks) being strictly consistent and more discriminating than another g (letter marks) can be shown in the figure 2(a).

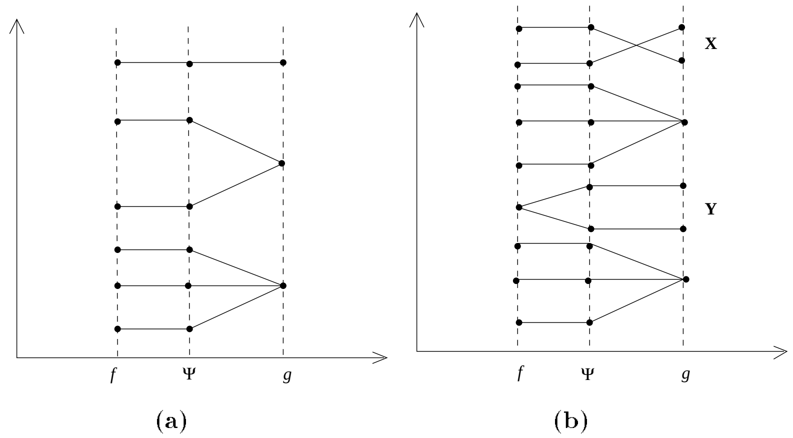

Fig. 2. Illustrations of two measures f and g. A link between two points indicates that the function values are the same on domain Ψ. In (a), f is strictly consistent and more discriminating than g. In (b), f is not strictly consistent or more discriminating than g. Counter examples on consistency (denoted by X in the figure) and discriminancy (denoted by Y) exist here

2.2 Statistical Consistency and Discriminancy of Two Measures

As we have already seen in the beginning of Section 2, counter examples on consistency and discriminancy do exist for AUC and accuracy. Therefore, it is impossible to prove the consistency and discriminancy on AUC and accuracy based on Definitions 1 and 2. What we will define and prove is the *probabilistic version* of the two definitions. Figure 2(b) illustrates a situation where one measure f is not completely consistent with g, and is not strictly more discriminating than g. In this case, we must consider the probability of being consistent and degree of being more discriminating. We extend the previous definitions on strict consistency and discriminancy to degree of consistency and degree of discriminancy, as follows:

Definition 3 (Degree of consistency) *For two measures f and g on domain Ψ, let $R = \{(a,b)|a,b \in \Psi, f(a) > f(b), g(a) > g(b)\}$, $S = \{(a,b)|a,b \in \Psi, f(a) > f(b), g(a) < g(b)\}$. The degree of consistency* [3] *of f and g is $\mathbf{C}$ ($0 \leq \mathbf{C} \leq 1$), where $\mathbf{C} = \frac{|R|}{|R|+|S|}$.*

Definition 4 (Degree of discriminancy) *For two measures f and g on domain Ψ, let $P = \{(a,b)|a,b \in \Psi, f(a) > f(b), g(a) = g(b)\}$, $Q = \{(a,b)|a,b \in \Psi, g(a) > g(b), f(a) = f(b)\}$. The degree of discriminancy for f over g is $\mathbf{D} = \frac{|P|}{|Q|}$.*

There are clear and important implications of these definitions of measures f and g in evaluating two machine learning algorithms, say A and B. If f and g are consistent to degree $\mathbf{C}$, then when f stipulates that A is better than B, there is a probability $\mathbf{C}$ that g will stipulate A is better than B. If f is $\mathbf{D}$ times more discriminating than g, then it is $\mathbf{D}$ times more likely that f can tell the difference between A and B but g cannot than that g can tell the difference between A and B but f cannot. Clearly, we require that $\mathbf{C} > 0.5$ and $\mathbf{D} > 1$ if we want to conclude a measure f is "better" than a measure g. This leads to the following definition:

Definition 5 *The measure f is* statistically consistent and more discriminating *than g if and only if $\mathbf{C} > 0.5$ and $\mathbf{D} > 1$. In this case, we say, intuitively, that f is a better measure than g.*

For the example of numerical and letter marks in the student evaluation discussed in Section 2.1, we can obtain that $\mathbf{C} = 1.0$ and $\mathbf{D} = \infty$, as the former is strictly consistent and more discriminating than the latter.

To prove AUC is statistically consistent and more discriminating than accuracy, we substitute f by AUC and g by accuracy in the definition above. To simplify our notation, we will use AUC to represent AUC values, and acc for accuracy. The domain Ψ is the ranked lists of testing examples (with n_0 positive and n_1 negative examples). Since we require $\mathbf{C} > 0.5$ and $\mathbf{D} > 1$ we will essentially need to prove:

[3] It is easy to prove that this definition is symmetric; that is, the degree of consistency of f and g is same as the degree of consistency of g and f.

Theorem 1. *Given a domain Ψ, let $R = \{(a,b)|AUC(a) > AUC(b), acc(a) > acc(b), a, b \in \Psi\}$, $S = \{(a,b)|AUC(a) < AUC(b), acc(a) > acc(b), a, b \in \Psi\}$. Then $\frac{|R|}{|R|+|S|} > 0.5$ or $|R| > |S|$.*

Theorem 2. *Given a domain Ψ, let $P = \{(a,b)|AUC(a) > AUC(b), acc(a) = acc(b), a, b \in \Psi\}$, $Q = \{(a,b)|acc(a) > acc(b), AUC(a) = AUC(b), a, b \in \Psi\}$. Then $|P| > |Q|$.*

We have proved these two theorems formally (submitted), and due to space limitation, the proofs are omitted.

2.3 Empirical Verification on AUC and Accuracy

We conduct experiments with two kinds of (artificial) testing sets to verify the statistical consistency and discriminancy between AUC and accuracy. The first kind is the balanced datasets with equal numbers of positive and negative examples, and the other kind is the unbalanced datasets with 25% positive and 75% negative examples. For the balanced dataset, we test ranked lists with 4, 6, 8, 10, 12, 14, and 16 examples. For the unbalanced dataset, we test ranked lists with 4, 8, 12, and 16 examples (so we can have exactly 25% of positive examples and 75% of negative examples). For each number of examples, we enumerate all possible ranked lists of positive and negative examples. For the balanced dataset with $2n$ examples, there are $\binom{2n}{n}$ such ranked lists.

We exhaustively compare all pairs of ranked lists to see how they satisfy the consistency and discriminating propositions probabilistically. To obtain degree of consistency, we count the number of pairs which satisfy "$AUC(a) > AUC(b)$ and $acc(a) > acc(b)$", and the number of pairs which satisfy "$AUC(a) > AUC(b)$ and $acc(a) < acc(b)$". We then calculate the percentage of those cases; that is, the degree of consistency. To obtain degree of discriminancy, we count the number of pairs which satisfy "$AUC(a) > AUC(b)$ and $acc(a) = acc(b)$", and the number of pairs which satisfy "$AUC(a) = AUC(b)$ and $acc(a) > acc(b)$".

Tables 5 and 6 show the experiment results for the balanced dataset. For consistency, we can see (Table 5) that for various numbers of balanced testing examples, given $AUC(a) > AUC(b)$, the number (and percentage) of cases that satisfy $acc(a) > acc(b)$ is much greater than those that satisfy $acc(a) < acc(b)$. When n increases, the degree of consistency (**C**) seems to approach 0.94, much larger than the required 0.5. For discriminancy, we can see clearly from Table 6 that the number of cases that satisfy $AUC(a) > AUC(b)$ and $acc(a) = acc(b)$ is much more (from 15.5 to 18.9 times more) than the number of cases that satisfy $acc(a) > acc(b)$ and $AUC(a) = AUC(b)$. When n increases, the degree of discriminancy (**D**) seems to approach 19, much larger than the required threshold 1.

Tables 7 and 8 show the experimental results for the unbalanced dataset (25% positive examples and 75% negative examples). We can draw similar conclusions that the degree of consistency (from 0.89 to 1.0) is much greater than 0.5, and

Table 5. Experimental results for verifying statistical consistency between AUC and accuracy for the balanced dataset

#	$AUC(a) > AUC(b)$ & $acc(a) > acc(b)$	$AUC(a) > AUC(b)$ & $acc(a) < acc(b)$	**C**
4	9	0	1.0
6	113	1	0.991
8	1459	34	0.977
10	19742	766	0.963
12	273600	13997	0.951
14	3864673	237303	0.942
16	55370122	3868959	0.935

Table 6. Experimental results for verifying AUC is statistically more discriminating than accuracy for the balanced dataset

#	$AUC(a) > AUC(b)$ & $acc(a) = acc(b)$	$acc(a) > acc(b)$ & $AUC(a) = AUC(b)$	**D**
4	5	0	NA
6	62	4	15.5
8	762	52	14.7
10	9416	618	15.2
12	120374	7369	16.3
14	1578566	89828	17.6
16	21161143	1121120	18.9

the degree of discriminancy (from 15.9 to 21.6) is certainly much greater than 1.0.

These empirical experiments show clearly that AUC is indeed a statistically consistent and more discriminating measure than accuracy for balanced or unbalanced datasets, as a measure for learning algorithms.

3 Comparison of Naive Bayes and Decision trees

We have established in Section 2 that AUC is a statistically consistent and more discriminating evaluation measure than accuracy. But most previous works only focussed on comparing the learning algorithms by accuracy. A well-accepted conclusion in the machine learning community is that the popular decision tree learning algorithm C4.5 [11] and Naive Bayes are very similar in predictive accuracy [8–10]. How do popular learning algorithms, such as decision trees and Naive Bayes, compare in terms of the better measure AUC? In this section, we will answer this question experimentally.

Table 7. Experimental results for verifying statistical consistency between AUC and accuracy for the unbalanced dataset

#	$AUC(a) > AUC(b)$ & $acc(a) > acc(b)$	$AUC(a) > AUC(b)$ & $acc(a) < acc(b)$	**C**
4	3	0	1.0
8	187	10	0.949
12	12716	1225	0.912
16	926884	114074	0.890

Table 8. Experimental results for verifying AUC is statistically more discriminating than accuracy for the unbalanced dataset

#	$AUC(a) > AUC(b)$ & $acc(a) = acc(b)$	$acc(a) > acc(b)$ & $AUC(a) = AUC(b)$	**D**
4	3	0	NA
8	159	10	15.9
12	8986	489	18.4
16	559751	25969	21.6

3.1 Representational Capacity

We first discuss some intuitions regarding the representational capacity of ranking in decision trees and Naive Bayes. In decision trees, the posterior probability of an example is the probability distribution of the leaf that the example falls into. Thus, all examples in the same leaf have the same probability, and they will be ranked randomly. This weakens substantially the capacity of decision trees in representing accurate ranking. This is because two contradictory factors are in play at the same time. On one hand, decision tree algorithms (such as ID3 and C4.5) aim at building a small decision tree. This results in more examples in the leaf nodes. Therefore, the many examples in the same leaves will be ranked randomly. In addition, a small tree implies a small number of leaves, and thus a small number of different probabilities. Thus, a small trees limits the discriminating power of the tree to rank examples. On the other hand, if the tree is large, the tree may not only overfit the data, but the number of examples falling into the leaf nodes becomes small, and thus the probability estimations of examples in the leaves would not be reliable. This would also produce poor ranking of testing examples.

This kind of contradiction does not exist in Bayesian networks. Naive Bayes calculates the posterior probability $p(c|e)$ based on $p(a_i|c)$, where a_i is the value of attribute A_i of example e with class c. Although Naive Bayes has only $2n+1$ parameters, the number of possible different posterior probabilities can be as many as 2^n. Therefore, intuitively speaking, even Naive Bayes has a significant advantage over decision trees in the capacity of representing different posterior probabilities.

3.2 Empirical Comparison

The fact that C4.5 produces poor probability estimations or AUC was recognized recently by some researchers [12]. Provost and Domingos [12] make the following improvements on C4.5 in an effort to improve its AUC scores:

1. **Turn off pruning.** C4.5 builds decision trees in two steps: building a large tree, and then pruning it to avoid the overfitting which results in a small tree with a higher predictive accuracy. However, Provost and Domingos show that pruning also reduces the quality of the probability estimation, as discussed above. For this reason, they choose to build the tree without pruning, resulting in substantially large trees.
2. **Smooth probability estimations by Laplace correction.** Because pruning has been turned off, the decision tree becomes large and has more leaves, and there are fewer examples falling into one leaf. The leaves with a small number of examples (e.g., 2) may produce probabilities of extreme values (e.g., 100%). In addition, it cannot provide reliable probability estimations. For this reason, Laplace correction was used to smooth the estimation and make it less extreme.

Provost and Domingos [12] called the resulting algorithm C4.4, and they showed that C4.4 produces decision trees with significantly better probability estimations than C4.5.

We conduct experiments to compare Naive Bayes, C4.5, and its recent improvement C4.4, using AUC as the evaluation criterion. We use 12 datasets with a relatively large number of examples from the UCI repository [13] as shown in Table 9. Our experiments follow the procedure below:

1. The continuous attributes in the dataset are discretized by the entropy-based method described in [14].
2. For each dataset, run Naive Bayes and C4.4 with the 5-fold cross-validation, and obtain the AUC on the testing set unused in the training.
3. Repeat Step 2 above 6 times and obtain an average AUC on the testing data.

Notice that we use the generalization of the AUC [6] for those datasets with more than two classes. Since Laplace correction has been used in C4.4 and significantly improves the AUC [12], we also use it in Naive Bayes.

The results are shown in Table 10. As we can see, in most datasets, Naive Bayes produces better AUC than C4.4, and C4.4 produces better AUC than C4.5 (as observed by [12]). We conduct a paired two-tailed t-test on the average AUC values of the 12 datasets, and conclude that Naive Bayes is significantly better than C4.4 in terms of AUC (with a 97.5% confidence level), and is certainly significantly better than C4.5 in terms of AUC (with a 99.6% confidence level). We also verify results in [12]: C4.4 is significantly better than C4.5 in terms of AUC (with a 98.0% confidence level).

Previous research have concluded that Naive Bayes and decision trees are very similar in prediction measured by accuracy [8–10]. As we have established

Table 9. Descriptions of the datasets used in our experiments

Dataset	Attributes	Class	Instances
Australia	14	2	690
breast	10	2	683
cars	6	4	700
dermatology	34	6	366
ecoli	7	8	336
hepatitis	4	2	320
import	24	2	205
iris	4	3	150
pima	8	2	392
segment	19	7	2310
vehicle	18	4	846
vote	16	2	232

in this paper, AUC is a better measure than accuracy. Our empirical comparisons between Naive Bayes and the decision tree algorithm C4.5 and its recent improvement C4.4 clearly show that Naive Bayes outperforms decision trees in terms of AUC.

Table 10. Experimental results of the AUC values of Naive Bayes, C4.4, and C4.5

Dataset	**NB**	**C4.4**	**C4.5**
Australia	75.8±0.8	73.1±0.1	72.0±0.1
breast	97.7±1.2	97.0±1.1	94.7±1.5
cars	93.6±0.3	94.3±0.0	90.3±0.1
dermatology	98.8±1.1	98.3±0.0	95.6±0.0
ecoli	99.3±0.1	99.2±0.0	98.5±0.0
hepatitis	62.1±1.1	59.5±0.1	60.8±0.1
import	99.2±0.2	100.0±0.0	100.0±0.0
iris	97.3±0.3	97.4±0.0	97.9±0.0
pima	77.6±0.9	75.6±0.1	75.7±0.1
segment	95.5±0.1	95.3±0.0	87.2±0.0
vehicle	91.3±0.9	88.1±0.1	84.3±0.1
vote	86.4±0.8	83.1±0.1	78.8±0.1
Average	**89.55**	**88.41**	**86.32**

4 Conclusions

In this paper, we argued that AUC is a better measure than accuracy based on formal definitions of discriminancy and consistency. Since AUC is applied under the same setting as accuracy — examples with only class labels — it allows us to re-evaluate many well-established conclusions based on accuracy in machine learning (such as the conclusion that decision trees, Naive Bayes, and neural networks are virtually equivalent in terms of accuracy). We have shown via extensive experimental comparisons that in terms of AUC, Naive Bayes predicts significantly better than the decision tree learning algorithms. Our results recommend AUC as a preferred "single number" evaluation measure over accuracy when evaluating and comparing classifiers. Many previous, well-accepted conclusions based on accuracy should also be re-evaluated using AUC. Such new conclusions will be very useful for machine learning and its applications (e.g., data mining for direct marketing).

Acknowledgements

We gratefully thank Foster Provost for kindly providing us with the source codes of C4.4, which is a great help to us in the comparison of C4.5 and C4.4 to other algorithms.

References

1. Ling, C.X., Zhang, H.: Toward Bayesian classifiers with accurate probabilities. In: Proceedings of the Sixth Pacific-Asia Conference on KDD (to appear). Springer (2002)
2. Cohen, W.W., Schapire, R.E., Singer, Y.: Learning to order things. Journal of Artificial Intelligence Research **10** (1999) 243–270
3. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: comparison under imprecise class and cost distribution. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. AAAI Press (1997) 43–48
4. Swets, J.: Measuring the accuracy of diagnostic systems. Science **240** (1988) 1285–1293
5. Provost, F., Fawcett, T., Kohavi, R.: The case against accuracy estimation for comparing induction algorithms. In: Proceedings of the Fifteenth International Conference on Machine Learning. Morgan Kaufmann (1998) 445–453
6. Hand, D.J., Till, R.J.: A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine Learning **45** (2001) 171–186
7. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition **30** (1997) 1145–1159
8. Kononenko, I.: Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. In Wielinga, B., ed.: Current Trends in Knowledge Acquisition. IOS Press (1990)
9. Langley, P., Iba, W., Thomas, K.: An analysis of Bayesian classifiers. In: Proceedings of the Tenth National Conference of Artificial Intelligence. AAAI Press (1992) 223–228
10. Domingos, P., Pazzani, M.: Beyond independence: conditions for the optimality of the simple Bayesian classifier. In: Proceedings of the Thirteenth International Conference on Machine Learning. (1996) 105 – 112

11. Quinlan, J.: C4.5: Programs for Machine Learning. Morgan Kaufmann: San Mateo, CA (1993)
12. Provost, F., Domingos, P.: Tree induction for probability-based ranking. Machine Learning (2003) To appear.
13. Merz, C., Murphy, P., Aha, D.: UCI repository of machine learning databases. In: Dept of ICS, University of California, Irvine. http://www.ics.uci.edu/ mlearn/MLRepository.html (1997)
14. Fayyad, U., Irani, K.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings of Thirteenth International Joint Conference on Artificial Intelligence. Morgan Kaufmann (1993) 1022–1027