Using AUC and Accuracy in Evaluating Learning Algorithms

Jin Huang Charles X. Ling
Department of Computer Science
The University of Western Ontario
London, Ontario, Canada N6A 5B7
{jhuang, cling}@csd.uwo.ca

Abstract

The area under the ROC (Receiver Operating Characteristics) curve, or simply AUC, has been recently proposed as an alternative single-number measure for evaluating the predictive ability of learning algorithms. However, no formal arguments were given as to why AUC should be preferred over accuracy. In this paper, we establish formal criteria for comparing two different measures for learning algorithms, and we show theoretically and empirically that AUC is, in general, a better measure (defined precisely) than accuracy. We then reevaluate well-established claims in machine learning based on accuracy using AUC, and obtain interesting and surprising new results. We also show that AUC is more directly associated with the net profit than accuracy in direct marketing, suggesting that learning algorithms should optimize AUC instead of accuracy in real-world applications. The conclusions drawn in this paper may make a significant impact to machine learning and data mining applications.

Note: *This paper integrates results in our papers published in IJCAI 2003 [22] and ICDM 2003 [15]. It also includes many new results. For example, the concept of indifferency in Section II-B is new, and Sections III-B, III-C, IV-A, IV-D, and V are all new and unpublished.*

Index Terms

Evaluation of learning algorithms, AUC vs accuracy, ROC

I. INTRODUCTION

The goal of classification learning algorithms is to build a classifier from a set of training examples with class labels such that the classifier can predict well the unseen testing examples. The predictive ability of the classification 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 0.51 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 data mining applications, however, accuracy is not enough. For example, in direct marketing, for example, we often need to promote the top X% (X can be 5 or 10) of customers during gradual roll-out, or we often deploy different promotion strategies to customers

with different likelihood of purchasing. 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 [20], 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 [7]. 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?

The ROC (Receiver Operating Characteristics) curve has been recently introduced to evaluate ranking performance of machine learning algorithms [27], [28]. Bradley [3] 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 [3]. However, no formal arguments or criteria have been established. Recently, other researchers have even used AUC to construct learning algorithms [11], [23]. But it is not clear if and why AUC is a better measure than accuracy. In general, 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, both empirically and formally, that AUC is indeed a statistically consistent and more discriminating measure than accuracy; that is, AUC is a better measure than accuracy.

One might ask why we need to care about anything more than accuracy, since by definition, classifiers only classify examples (and do not care about ranking and probability). We can answer this question from three aspects. First, as we discussed earlier, even with labelled training and testing examples, most classifiers do produce probability estimations that can rank training/testing examples. Ranking is very important in most real-world applications. As we will see in Section II-A that AUC directly measures ranking, we should choose classifiers with better AUC, thus producing better ranking. Second, and more importantly, if we build classifiers that optimize AUC (instead of accuracy), such classifiers produce not only better AUC (a natural consequence), but

also better accuracy (a surprising result), compared to classifiers that only optimize the accuracy [23]. To make an analogy, when we train workers on a more complex task, they will do better on a simple task than workers who are trained only on the simple task. Third, we will show (Section V) that AUC is associated more directly with the net profit in direct marketing than accuracy. Thus optimizing AUC improves the ROI (Return of Investment) in real-world applications.

Our work is quite significant for several reasons. First, we establish rigourously, in [22], that even given only labelled examples, AUC is a better measure (defined in Section II-B) than accuracy. Our result suggests that AUC should replace accuracy in comparing learning algorithms in the future. Second, 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 [17], [18], [9]. Since we will establish that AUC is a better measure, are those learning algorithms still very similar as measured by AUC? How does recent Support Vector Machine (SVM) [2], [8], [30] compare to traditional learning algorithms such as Naive Bayes and decision trees in accuracy and AUC? We perform extensive experimental comparisons to compare Naive Bayes, decision trees, and SVM to answer these questions in Section IV. Third, we show that AUC is more directly associated with the net profit in direct marketing than accuracy (Section V). This suggests that in real-world applications of machine learning and data mining, we should use learning algorithms optimizing AUC instead of accuracy. Most learning algorithms today still optimize accuracy directly (or indirectly through entropy, for example) as their goals. Our conclusions may make significant impacts in data mining research and applications.

II. Criteria for Comparing Evaluation Measures

We start with some intuitions in comparing AUC and accuracy, and then we present formal definitions in comparing evaluation measures for learning algorithms.

A. AUC vs Accuracy

Hand and Till [12] present a simple approach to calculating the AUC of a classifier below.

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

TABLE I

AN EXAMPLE FOR CALCULATING AUC WITH r_i

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

TABLE II

AN EXAMPLE IN WHICH TWO CLASSIFIERS HAVE THE SAME CLASSIFICATION ACCURACY, BUT DIFFERENT AUC VALUES

Classifier 1	+	|	+	+	+	+
Classifier 2	+	|	+	+	+	+

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 I 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 I is $\frac{(5+7+8+9+10) \quad 5\times 6/2}{5\times 5}$, which is 24/25. It is clear that AUC obtained by Equation 1 is a way to measure 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 shown to be equivalent to the Wilcoxon statistic rank test [3].

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 II.

Clearly, both classifiers produce an accuracy of 80% (or an error rate of 20% with one false positive and one false negative), and thus the two classifiers are equivalent in terms of accuracy. 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 I), 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 III 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 accuracy of Classifier 3 is 60%, while the accuracy of Classifier 4 is 80% (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 higher accuracy; that is, AUC and accuracy sometimes contradict to each other. Therefore, which one should we really "trust"?

TABLE III

A COUNTER EXAMPLE IN WHICH ONE CLASSIFIER HAS HIGHER AUC BUT LOWER CLASSIFICATION ACCURACY

Classifier 3			+	+	|			+	+	+
Classifier 4	+				|	+	+	+	+	

Another intuitive argument for AUC against 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 $n_0 n_1 + 1$ different AUC values ($0/n_0 n_1$, $1/n_0 n_1$, ..., $n_0 n_1/n_0 n_1$), generally more than $n+1$. However, counter examples also exist in this regard. Table IV illustrates two classifiers with the same AUC but different accuracies. Here, we see that both 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. The two measures may have nothing to do with each other.

TABLE IV

A COUNTER EXAMPLE IN WHICH TWO CLASSIFIERS HAVE SAME AUC BUT DIFFERENT CLASSIFICATION ACCURACIES

Classifier 5			+	+		|	+	+			+
Classifier 6			+	+	+	|			+		+

Last, there exist cases where neither AUC nor accuracy can tell the difference. Figure V shows such an example. The AUC of both classifiers is $\frac{3}{5}$, and the accuracy of both classifier is 60%. If two measures have too many such indistinguishable cases, they would not be considered desirable. How often does this happen for AUC and accuracy?

TABLE V

AN EXAMPLE IN WHICH TWO CLASSIFIERS HAVE SAME AUC AND SAME CLASSIFICATION ACCURACY

Classifier 7	+	+	|	+	+			+
Classifier 8	+	+	|	+		+	+	

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

B. (Strict) Consistency and Discriminancy

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, but not vise versa.[1]

This intuitive meaning of consistency and discriminancy can be made precise as the following definitions. We assume that Ψ is the domain of two functions f and g which return values as some performance measure.

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 $f(a) \neq f(b)$ and $g(a) = g(b)$, and there exist no a, $b \in \Psi$ such that $g(a) \neq g(b)$ 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 (A), but it is not possible to have students

[1]As we have already seen in Section II-A, counter examples on strict consistency and discriminancy do exist for AUC and accuracy. See Section II-C for definitions on *statistical* consistency and discriminancy between two measures.

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 1(a).

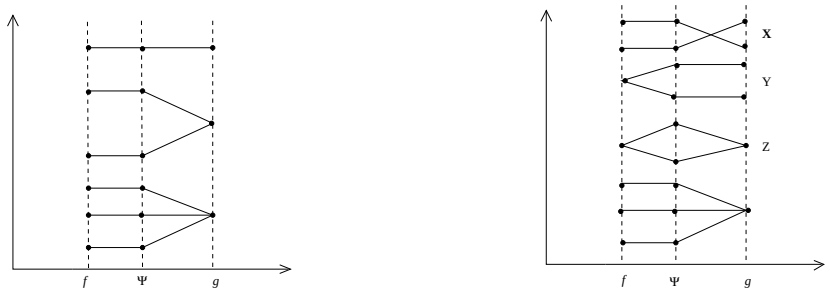

Fig. 1. Illustrations of two measures f and g. In (a), f is strictly consistent and more discriminating than g. In (b), f is not strictly consistent or more discriminating than g. Counter examples on consistency (denoted by X in the figure), discriminancy (denoted by Y), and indifferency (denoted by Z) exist here

C. Statistical Consistency and Discriminancy

As we have already seen in Section II-A, counter examples on consistency (Table III) and discriminancy (Table IV) 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. Figure 1(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. What we will define and prove is the *probabilistic version* of the two definitions on strict consistency and discriminancy. That is, we extend the previous definitions 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 [2] of f and g is $\mathbf{C}$ ($0 \leq \mathbf{C} \leq 1$), where $\mathbf{C} = \frac{|R|}{|R|+|S|}$.

[2]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.

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 $\mathbf{D} = \frac{|P|}{|Q|}$.

As we have seen in Table V and Figure 1b, there may exist cases where the two measures cannot tell the difference. The frequency of such cases is the Degree of Indifferency defined below.

Definition 5 (Degree of Indifferency): For two measures f and g on domain Ψ, let $V = \{(a,b)|a,b \in \Psi, a \neq b, f(a) = f(b), g(a) = g(b)\}$, $U = \{(a,b)|a,b \in \Psi, a \neq b\}$. The degree of indifferency for f and g is $\mathbf{E} = \frac{|V|}{|U|}$.

We would naturally require $\mathbf{E} \neq 1$ (or $\mathbf{E} < 1$), but this is true for almost all useful measures. For $\mathbf{E} = 1$ to happen, the measures must return the same values for all elements in the domain. That is, if one measure always returns a constant (such as 60%), and the other measure also always returns a constant (such as 80%), then $\mathbf{E} = 1$. Therefore, we will omit the requirement on $\mathbf{E}$ in the rest of the discussion.

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 agree (stipulating 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 6: 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.

The statistical consistency and discriminancy is a special case of the strict consistency and more discriminancy. For the example of numerical and letter marks in the student evaluation discussed in Section II-B, 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 ranked lists of testing

examples.

We have proven the following two theorems under the condition that Ψ contains all possible balanced binary ranked lists in [22]. That is, AUC is indeed statistically consistent and more discriminating than accuracy if the domain contains all possible binary, balanced (with the same number of positive and negative examples) ranked lists.

Theorem 1: Given a domain Ψ of all possible balanced binary ranked lists, 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 of all possible balanced binary ranked lists, 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 $|P| > |Q|$.

See [22] for the proof.

III. Empirical Verification on AUC and Accuracy

In this section we present an empirical verification of the two theorems on artificial datasets. This is necessary for two reasons. First, as we have only been able to prove the theorems with certain limitations (e.g., binary, balanced datasets), we also want to know if the theorems are true with imbalanced and multi-class datasets. Most real-world datasets are imbalanced with multiple class values. More importantly, empirical evaluations on artificial datasets will give us intuitions on the ranges of the degree of consistency $\mathbf{C}$, the degree of discriminancy $\mathbf{D}$, and degree of indifferency $\mathbf{E}$, on different types of datasets (balanced, imbalanced, and multi-class datasets).

A. Balanced Binary Data

Even though we have proved that AUC is indeed statistically consistent and more discriminating than accuracy if the domain contains all possible binary, balanced ranked lists [22], we still perform an empirical evaluation in order to gain an intuition on the ranges of the degree of consistency $\mathbf{C}$, the degree of discriminancy $\mathbf{D}$, and degree of indifferency $\mathbf{E}$.

Thus, the datasets in this experimental setting are balanced with equal numbers of positive and negative examples (binary class). We test datasets with 4, 6, 8, 10, 12, 14, and 16 testing

examples. For each case, we enumerate all possible ranked lists of (equal numbers of) positive and negative examples. For the 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 VI and VII show the experiment results. For consistency, we can see (Table VI) 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 0.93, much larger than the required 0.5. For discriminancy, we can see clearly from Table VII 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.

TABLE VI

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 VII

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	∞
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

These experimental results verify empirically that AUC is indeed a statistically consistent and more discriminating measure than accuracy for the balanced binary datasets.

We also obtain the degree of indifferency between AUC and accuracy for the balanced binary datasets. The results can be found in Table VIII. As we can see, the degree of indifferency **E** is very small: from about 7% to 2%, and the trend is decreasing as the number of examples increases. This is desirable as for most cases (with a probability $1 \quad E$), AUC and accuracy are not indifferent; that is, they are either consistent, inconsistent, or one is more discriminant than another.

B. Imbalanced Datasets

We extend our previous results on the balanced datasets with binary classes to imbalanced datasets and datasets with multiple classes. We will experimentally verify that statistical consistency and discriminancy still hold in these relaxed conditions.

We first test imbalanced binary datasets, which have 25% positive and 75% negative examples. We use ranked lists with 4, 8, 12, and 16 examples (so we can have exactly 25% of positive examples and 75% of negative examples). We still use the same formula 1 to calculate AUC, but for accuracy, we must decide the cut-off point. We make a reasonable assumption that the class distributions of training sets and testing sets are the same. That is, the cut-off point of the ranked list is at the 75% position: the lower 75% of the ranked testing examples are classified

TABLE VIII

EXPERIMENTAL RESULTS FOR THE DEGREE OF INDIFFERENCY BETWEEN AUC AND ACCURACY FOR THE BALANCED BINARY DATASET.

#	$AUC(a) = AUC(b)$ & $acc(a) = acc(b)$	(a, b) & $a \neq b$	**E**
4	1	15	0.067
6	10	190	0.053
8	108	2415	0.045
10	1084	31626	0.034
12	11086	426426	0.026
14	117226	5887596	0.020
16	1290671	82812015	0.016

as negative, and the top 25% of the ranked testing examples are classified as positive. Tables IX and X show the experimental results for the imbalanced datasets (with 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 the degree of discriminancy (from 15.9 to 21.6) is certainly much greater than 1.0. However, compared to the results in balanced datasets (Tables VI and VII), we can see that degree of consistency is lowered but the degree of discriminancy is higher when datasets are imbalanced.

TABLE IX

EXPERIMENTAL RESULTS FOR VERIFYING STATISTICAL CONSISTENCY BETWEEN AUC AND ACCURACY FOR THE IMBALANCED BINARY 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

We have also obtained the degree of indifferency for the imbalanced binary datasets as shown in Table XI. Compared to the results in Table VIII, we can conclude that the degree

TABLE X

EXPERIMENTAL RESULTS FOR VERIFYING AUC IS STATISTICALLY MORE DISCRIMINATING THAN ACCURACY FOR THE IMBALANCED BINARY 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

of indifferency is basically the same.

TABLE XI

EXPERIMENTAL RESULTS FOR THE DEGREE OF INDIFFERENCY BETWEEN AUC AND ACCURACY FOR THE IMBALANCED BINARY DATASET.

#	$AUC(a) = AUC(b)$ & $acc(a) = acc(b)$	(a, b) & $a \neq b$	**E**
4	0	6	0
8	12	378	0.032
12	629	24090	0.026
16	28612	1655290	0.017

To see the effect of the class distribution to degree of imbalanced consistency and discriminancy, we fix the number of the testing examples to 10, and vary the number of positive examples as 5 (balanced), 6, 7, 8, and 9. Table XII shows the changes of consistency and discriminancy with different class distribution. As we can see, except for the extreme cases at the two ends, the more imbalanced the class distribution, the lower the degree of consistency (but still well above 0.5), and the higher the degree of discriminancy. These results are very interesting as they provide intuitions on degree of consistency and discriminancy in the binary datasets with different class distributions.

TABLE XII

EXPERIMENTAL RESULTS FOR SHOWING THE VARIATION OF DEGREE OF CONSISTENCY AND DISCRIMINANCY WITH DIFFERENT CLASS DISTRIBUTION FOR BINARY DATASET

n_0	n_1	**C**	**D**
1	9	1.0	∞
2	8	0.926	22.3
3	7	0.939	15.5
4	6	0.956	14.9
5	5	0.963	15.2
6	4	0.956	14.9
7	3	0.939	15.5
8	2	0.926	22.3
9	1	1.0	∞

C. Multiclass Datasets

This set of experiments concerns with artificial datasets with multiple classes (balanced only). Some complexity enters the scene as we try to enumerate all possible ranked lists for multiple classes. In the binary cases, the lists are always ranked according to the probability of positive, which is the same as reversely to the probability of negative. It is much more complicated for multiple classes cases. Here each class can be ranked separately with different results. To do so, we actually need to generate (or simulate) probabilities of multiple classes. More specifically, for each testing ranked list with c classes, the class distribution of each example is randomly generated (but sum of all class probabilities is 1). The class with the largest probability is the "correct" class. We make sure that there is an equal number of examples in each class. We generate a large number of such lists (it is impossible to enumerate all such lists with different class probability distributions as the number is infinite), and we then randomly choose two lists from the large pool of such lists to calculate the relation between their AUC and accuracy values. We do that 50,000 times from a large pool to get an averaged degree of consistency and discriminancy to approximate all possible ranked lists with the uniform distribution.

The actual calculation of AUC and accuracy also needs to be extended. For AUC calculation for multiple classes, we use a simple generalization proposed in [12], as follows: Recall that each example has a label indicating the class it actually belongs to. For a ranked list of c classes, each

example is assigned with c probabilities ($p_1, p_2, \cdots, p_c$) for its c classes. For all the examples with class labels i and j, we first sort them incrementally by the probability p_i value, and we calculate the AUC value as $AUC(i,j)$. Then we sort them incrementally by the probability p_j value, and we calculate the AUC value as $AUC(j,i)$. The AUC between classes i and j is $\hat{AUC}(i,j) = \frac{AUC(i,j)+AUC(j,i)}{2}$. The AUC of this ranked list is the average AUC values for every two classes, which is $\frac{2}{c(c-1)} \sum_{i<j} \hat{AUC}(i,j)$.

For accuracy calculation, we use the same assumption that the class distribution in the testing set is the same. Therefore, the list of examples is partitioned into c consecutive portions, and each portion is assigned as one of the c classes. This assumption is not restrictive as any ranked list is a permutation of this one.

Table XIII shows the experimental results for the consistency and discriminancy of the multiclass datasets. We can clearly see that when the number of classes increases, the degree of consistency is decreasing (the trend suggests that the rate of decreasing does slow down), while the degree of discriminancy increases. We have not experimented with imbalanced multiclass datasets. The conclusions of previous experiments can very likely be extended: the further imbalanced the datasets, the lower the degree of consistency and the higher the degree of discriminancy.

TABLE XIII

EXPERIMENTAL RESULTS FOR VERIFYING CONSISTENCY AND DISCRIMINANCY BETWEEN AUC AND ACCURACY FOR MULTICLASS. THE NUMBER OF CLASSES RANGES FROM 3 TO 10, AND THERE ARE 2 EXAMPLES FOR EACH CLASS.

# of class	C	D
3	0.897	5.5
4	0.828	7.1
5	0.785	9.5
6	0.757	12.1
7	0.736	15.0
8	0.721	18.3
9	0.705	21.6
10	0.696	25.3

To conclude, for both balanced or imbalanced, binary or multiclass datasets, our experiments suggest that AUC is statistically consistent with accuracy ($\mathbf{C} > 0.5$), and AUC is statistically

more discriminant than accuracy ($\mathbf{D} > 1$); that is, AUC is a better measure than accuracy.

IV. Comparing Naive Bayes, Decision Trees, and SVM

We have established, empirically (Section III) and formally [22], that AUC is a statistically consistent and more discriminating evaluation measure than accuracy on artificial datasets. It would be interesting to find out if this is also true for the real-world datasets. In Section IV-D we will empirically verify this with benchmark datasets from the UCI repository [1].

More importantly, most previous work only focussed on comparing the learning algorithms in accuracy. A well-accepted conclusion in the machine learning community is that the popular decision tree learning algorithm C4.5 [29] and Naive Bayes are very similar in predictive accuracy [17], [18], [9]. How do popular learning algorithms, such as decision trees and Naive Bayes, compare in terms of the better measure AUC? How does recent Support Vector Machine (SVM) compare to traditional learning algorithms such as Naive Bayes and decision trees? We attempt to answer these questions in Sections IV-B and IV-C.

A. Representational Capacity

We first discuss some intuitions regarding the representational capacity of ranking in decision trees and Naive Bayes. For decision trees, the posterior probability of an example is the probability distribution of the leaf which 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 (see [21] for an improvement on more accurate probability predictions in decision trees). 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.

B. Comparing Naive Bayes and Decision Trees

The popular decision tree learning algorithm C4.5 have been recently observed to produce poor probability estimations on AUC [31], [28], [26]. Several improvements have been proposed, and we want to include a recent improvement, C4.4 [26], in our comparison.

Provost and Domingos [26] 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.

They called the resulting algorithm C4.4, and showed that C4.4 produces decision trees with significantly higher AUC than C4.5 [26].

We conduct our experiments to compare Naive Bayes, C4.5, and its recent improvement C4.4, using both accuracy and AUC as the evaluation criterion. We use 18 datasets (both binary and multi-class) with a relatively large number of examples from the UCI repository [1]. SVM is not involved in the comparison as some datasets are multiple classes. See Section IV-C for reasons.

Our experiments follow the procedure below:

1) The continuous attributes in all datasets are discretized by the entropy-based method described in [10].
2) For each dataset, create 10 pairs of training and testing sets with 10-fold cross-validation, and run Naive Bayes, C4.5, and C4.4 on the *same* training sets and test them on the *same* testing sets to obtain the testing accuracy and AUC scores.

TABLE XIV

DESCRIPTIONS OF THE DATASETS USED IN OUR EXPERIMENTS

Dataset	Attributes	Class	Instances
breast	9	2	683
cars	6	2	700
credit	15	2	653
dermatology	34	4	366
echocardio	4	2	61
eco	6	2	332
glass	8	6	214
heart	8	2	261
hepatitis	8	2	112
import	23	2	205
iris	4	3	150
liver	2	2	345
mushroom	21	2	8124
pima	6	2	392
solar	12	6	1066
thyroid	24	2	2000
voting	16	2	232
wine	13	3	178

The averaged results on accuracy are shown in Table XV, and on AUC in Table XVI. As we can see from Table XV, the three algorithms have very similar predictive accuracy. The two tailed, paired t-test with 95% confidence level (same for other t-tests in the rest of the paper) shows that there is no statistical difference in accuracy between Naive Bayes and C4.4, Naive Bayes and C4.5, and C4.4 and C4.5. This verifies results of previous publications [17], [18], [9].

When we analyze the table for AUC (Table XVI), we get some very interesting results. The

TABLE XV

PREDICTIVE ACCURACY VALUES OF NAIVE BAYES, C4.4, AND C4.5

Dataset	**NB**	**C4.4**	**C4.5**
breast	97.5±2.9	92.9±3.0	92.8±1.2
cars	86.4±3.7	88.9±4.0	85.1±3.8
credit	85.8±3.0	88.1±2.8	88.8±3.1
dermatology	98.4±1.9	94.0±3.5	94.0±4.2
echocardio	71.9±1.8	73.6±1.8	73.6±1.8
ecoli	96.7±2.2	96.4±3.1	95.5±3.9
glass	71.8±2.4	73.3±3.9	73.3±3.0
heart	80.8±7.3	78.9±7.6	81.2±5.6
hepatitis	83.0±6.2	81.3±4.4	84.02±4.0
import	96.1±3.9	100.0±0.0	100.0±0.0
iris	95.3±4.5	95.3±4.5	95.3±4.5
liver	62.3±5.7	60.5±4.8	61.1±4.9
mushroom	97.2±0.8	100.0±0.0	100.0±0.0
pima	71.4±5.8	71.9±7.1	71.7±6.8
solar	74.0±3.2	73.0±3.1	73.9±2.1
thyroid	95.7±1.1	96.0±1.1	96.6±1.1
voting	91.4±5.6	95.7±4.6	96.6±3.9
wine	98.9±2.4	95.0±4.9	95.5±5.1
Average	**86.4**	**86.4**	**86.6**

average predictive AUC score of Naive Bayes is slightly higher than that of C4.4, and much higher than that of C4.5. The paired t-test shows that the difference between Naive Bayes and C4.4 is not significant, but the difference between Naive Bayes and C4.5 is significant. (The difference between C4.4 and C4.5 is also significant, as observed by [26]). That is, Naive Bayes outperforms C4.5 in AUC with significant difference.

This conclusion is quite significant to the machine learning and data mining community. Previous research concluded that Naive Bayes and C4.5 are very similar in prediction measured by accuracy [17], [18], [9]. As we have established in this paper, AUC is a better measure than accuracy. Further, our results show that Naive Bayes and C4.4 outperform the most popular

decision tree algorithm C4.5 in terms of AUC. This indicates that Naive Bayes (and C4.4) should be favoured over C4.5 in machine learning and data mining applications, especially when ranking is important.

TABLE XVI

PREDICTIVE AUC VALUES OF NAIVE BAYES, C4.4, AND C4.5

Dataset	NB	C4.4	C4.5
breast	97.5±0.9	96.9±0.9	95.1±2.4
cars	92.8±3.3	94.1±3.2	91.4±3.5
credit	91.9±3.0	90.4±3.2	88.0±4.1
dermatology	98.6±0.1	97.5±1.1	94.6±3.3
echocardio	63.8±2.1	69.4±2.2	68.9±2.3
ecoli	97.0±1.1	97.0±1.0	94.3±3.6
glass	76.1±2.4	73.1±2.6	71.3±3.3
heart	82.7±6.1	80.1±7.8	76.2±7.0
hepatitis	76.5±4.4	62.9±8.2	59.2±6.8
import	91.7±4.5	94.4±2.0	95.1±2.6
iris	94.2±3.4	91.8±3.8	92.4±4.6
liver	61.5±5.9	59.6±5.7	60.5±5.0
mushroom	99.7±0.1	99.9±0.0	99.9±0.0
pima	75.9±4.2	73.4±7.3	72.4±7.4
solar	88.7±1.7	87.7±1.9	85.2±2.8
thyroid	94.9±1.8	94.3±2.6	92.1±5.5
voting	91.4±3.7	95.2±2.2	93.4±3.7
wine	95.3±1.8	94.4±1.2	91.6±4.0
Average	**87.2**	**86.2**	**84.5**

C. Comparing Naive Bayes, Decision Trees, and SVM

In this section we compare accuracy and AUC of Naive Bayes, C4.4, and C4.5 to the recently developed SVM [33], [8], [5] on the datasets from the UCI repository. Such an extensive comparison with a large number of benchmark datasets is still rare [25]; most previous work (such as [13]) limited to only a few comparisons, with the exception of [25]. SVM is essentially

a binary classifier, and although extensions have been made to multiclass classification [32], [14] there is no consensus which is the best. Therefore, we take the 13 binary-class datasets from the 18 datasets in the experiments involving SVM. [25] also only used binary datasets for the classification for the same reason.

For SVM we use the software package LIBSVM [6] modified to directly output the evaluation of the hyperplane target function as scores for ranking. We used the Gaussian Kernel for all the experiments. The parameters C (penalty for misclassification) and gamma (function of the deviation of the Gaussian Kernel) were determined by searching for the maximum accuracy in the two-dimensional grid formed by different values of C and gamma in the 3-fold cross-validation on the training set (so the testing set in the original 10-fold cross-validation is not used in tuning SVM). C was sampled at 2^{-5}, 2^{-3}, 2^{-1}, ..., 2^{15}, and gamma at 2^{-15}, 2^{-13}, 2^{-11}, ..., 2^{3}. Other parameters are set default values by the software. This experiment setting is similar to the one used in [25]. The experiment procedure is the same as discussed earlier.

The predictive accuracy and AUC of SVM on the testing sets of the 13 binary datasets are listed in Table XVII. As we can see, the average predictive accuracy of SVM on the 13 binary datasets is 87.8%, and the average predictive AUC is 86.0%. From Table XV we can obtain the average predictive accuracy of Naive Bayes, C4.4, and C4.5 on the 13 binary datasets is 85.9%, 86.5%, and 86.7%, respectively. Similarly, from Table XVI we can obtain the average predictive AUC of Naive Bayes, C4.4, and C4.5 on the 13 binary datasets is 86.0%, 85.2%, and 83.6%, respectively.

Several interesting conclusions can be drawn. First, the average predictive accuracy of SVM is slightly higher than other algorithms in comparison. However, the paired t-test shows that the difference is *not* statistically significant. Secondly, the average predictive AUC scores showed that SVM, Naive Bayes, and C4.4 are very similar. In fact, there is no statistical difference among them. However, SVM does have significantly higher AUC than C4.5, so does Naive Bayes and C4.4 (as observed in the early comparison in Section IV-B). Our results on SVM may be inconsistent with some other comparisons involving SVM which showed superiority of SVM over other learning algorithms [25], [19], [4], [16] We think that one major difference is in data pre-processing: we have discretized all numerical attributes (see Section IV-B) as Naive Bayes requires all attributes to be discrete. Discretization is also an important pre-processing step in data mining [24]. The discretized attributes are named 1, 2, 3, and so on. Decision

trees and Naive Bayes then take discrete attributes directly. For SVM, those values are taken as numerical attributes after normalization. In most previous comparisons, numerical attributes are used directly in SVM. However, we think that our comparisons are still fair since all algorithms use the same training and testing datasets after discretization. If there is loss of information during discretization, the decision trees, Naive Bayes, and SVM would suffer equally from it. The other difference is that we did not seek for problem-specific, best kernels for SVM. This is fair as Naive Bayes, C4.5, and C4.4, are run automatically in the default, problem-independent parameter settings.

TABLE XVII

PREDICTIVE ACCURACY AND AUC OF SVM ON THE 13 BINARY DATASETS

Dataset	**Accuracy**	**AUC**
breast	96.5±2.3	97.3±1.3
cars	97.0±1.3	98.6±0.4
credit	86.4±2.9	90.4±3.0
echocardio	73.6±1.8	71.5±2.0
ecoli	96.4±3.1	95.0±2.8
heart	79.7±8.2	82.1±8.3
hepatitis	85.8±4.2	64.2±8.7
import	100.0±0.0	93.8±0.6
liver	60.5±4.8	61.6±5.6
mushroom	99.9±0.1	99.9±0.0
pima	72.2±6.3	72.2±7.5
thyroid	96.7±1.3	95.8±3.3
voting	97.0±3.5	95.3±0.7
Average	**87.8**	**86.0**

D. AUC and Accuracy on Real-World Datasets

We have established, empirically (Section III) and formally [22], that AUC is a statistically consistent and more discriminating evaluation measure than accuracy on artificial datasets. It would be interesting to find out if this is also true for the real-world datasets.

To verify statistical consistency and discriminancy between accuracy and AUC on real-world datasets with imbalanced and multiple class distributions, we compare every pair (C4.4 vs Naive

Bayes, C4.5 vs Naive Bayes, and C4.5 vs C4.4) of the learning algorithms in the 18 datasets from Table XV and Table XVI. To obtain finer results, we actually compare pairs of learning algorithms on each cross-validation test set (there are a total of 180 such testing sets from 18 datasets with 10-fold cross validation). For each pair of algorithms, we do not care which one is better (this has been answered in Section IV-B); instead, we only care if the two algorithms are consistent or not in AUC and accuracy, and if one measure is more discriminant than another. The results are reported in Table XVIII. In the table left column, + means, in the "algorithm A vs algorithm B" comparison, A is better than B, means A is worse than B, = means A is the same as B, and $\neq$ means A is not the same as B (in the paired t-test). Thus, the number 84 in Table XVIII means that there are 84 cross-validation testing sets (among 180) in which C4.4 is better than Naive Bayes in both accuracy and AUC, or C4.4 is worse than Naive Bayes in both accuracy and AUC. That is, C4.4 and Naive Bayes are consistent in both accuracy and AUC on 84 cross-validation testing sets. The number 29 in the table means that there are 29 cross-validation test sets (among 180) in which C4.4 is better than Naive Bayes in accuracy but worse in AUC, or C4.4 is worse than Naive Bayes in accuracy but better in AUC. That is, C4.4 and Naive Bayes are inconsistent in accuracy and AUC on 29 cross-validation testing sets. The ratio of 84/(84+29)=0.743 is then the degree of consistency **C**. [3] Similarly, the numbers in the row "acc=/AUC$\neq$" indicates the number of cross-validation testing sets that the two algorithms are same in accuracy but different in AUC, and "acc$\neq$/AUC=" indicates the number of cross-validation testing sets that the two algorithms are different in accuracy but same in AUC. The ratio of the two numbers (for example, 55/2=27.5) is then the estimated degree of discriminancy **D**. From the estimated values of **C** and **D** in Table XVIII, we can clearly see that for all pairs of the algorithms compared over 180 cross-validation testing sets, they are statistically consistent ($\mathbf{C} > 0.5$), and AUC is more discriminant than accuracy ($\mathbf{D} > 1$).

We can also see that the degree of indifferency of C4.5 vs C4.4 (0.172) is higher than C4.5 vs NB (0.105), and is higher than C4.4 vs NB (0.056). This indicates that C4.5 and C4.4 produce more similar results (ranked lists) than the other pairs (if two algorithms predict exactly the same, they will be indifferent by any measure). This is somewhat expected as C4.4 is an improved

[3]The definitions, theorems, and empirical verifications of consistency discussed previously are based on the domain which contains all possible ranked list (or datasets) with a uniform distribution. Here the domain is the datasets used in the experiments.

version of C4.5, so it would produce similar ranked lists as C4.5.

Last, we can see that the degree of discriminancy of C4.5 vs C4.4 (67) is larger than C4.5 vs NB (46), and is larger than C4.4 vs NB (27.5). This indicates, intuitively, that the difference between AUC and accuracy is more evident in the former ones. Indeed, C4.4 and Naive Bayes are more close in their prediction in AUC (see Table XVI), and thus, they are more similar in the effect of AUC and accuracy on the testing datasets.

TABLE XVIII

THE CONSISTENCY AND DISCRIMINANCY OF ACCURACY AND AUC FOR PAIRS OF LEARNING ALGORITHMS

	C4.4 vs. NB	**C4.5 vs. NB**	**C4.5 vs. C4.4**
acc+/AUC+ or acc /AUC	84	83	45
acc+/AUC or acc /AUC+	29	31	36
Degree of consistency **C**	0.743	0.728	0.556
acc=/AUC≠	55	46	67
acc≠/AUC=	2	1	1
Degree of discriminancy **D**	27.5	46	67
acc=/AUC=	10	19	31
Degree of indifferency **E**	0.056	0.106	0.172

E. Summary

To summarize, our extensive experiments in this section allows us to draw the following conclusions:

- The comparisons between pairs of learning algorithms (Section IV-D) verify the theorems in Section II-C on real-world datasets. That is, AUC and accuracy are statistically consistent, and AUC is more statistically discriminant than accuracy on real-world datasets.
- The average predictive accuracy of the four learning algorithms compared (Naive Bayes, C4.5, C4.4, and SVM) are very similar. There is no statistical difference between them. The recent SVM does produce slightly higher average accuracy but the difference on the 13 binary datasets is not statistically significant (Sections IV-B and IV-C).

- The average predictive AUC values of Naive Bayes, C4.4, and SVM are very similar (no statistical difference), and they are all higher with significant difference than C4.5 (Sections IV-B and IV-C).

Our conclusions will provide important guidelines in data mining applications on real-world datasets.

V. Optimizing Profit with AUC and Accuracy

We have compared theoretically and empirically the two measures, AUC and accuracy, for machine learning algorithms, and showed that AUC is a better measure than accuracy. However, in real-world applications, neither AUC nor accuracy is the final goal. The final goal of using machine learning (data mining) is to optimize some sort of profit measure.[4] For example, banks and insurance companies may have large database of customers to whom they want to sell certain products. They may use machine learning (data mining) software to identify desirable customers (by maximizing accuracy or AUC), but the final evaluation is if the predictions make profit to the companies. In this section, we will study the effect of improving profit in terms of increasing AUC or accuracy in a simulated direct marketing campaign.

Assume that customers in the database are described by a number of attributes, and each customer is potentially either a buyer or non-buyer of a certain product. As this is a binary classification problem, all the potential buyers are assumed to make an equal amount of revenue if they are approached by the campaign. Assume that there is a cost associated with the promotion to each customer, and that such a cost is also constant. Then if the company approaches all customers predicted as buyers, and ignores all customers predicted as non-buyers, the final net profit will be proportional to the number of customers who are correctly predicted. That is, the final net profit is equivalent to the accuracy. In this case, optimizing accuracy would indeed optimize the net profit.

In the real world applications, however, the company may only want to promote a small percentage of the top likely buyers predicted, different buyers may bring in different revenue to the company, and the promotional cost may be different for customers (i.e., the company

[4]However, because such a profit measure is hard to define within learning algorithms, we often optimize measures such as accuracy or AUC to optimize the profit indirectly.

would spend more money in promoting highly likely buyers). In this study, we assume that the company only wants to promote a small percentage of the top likely buyers, as the revenue and cost of promotion are unknown in the datasets.

As we have discussed in Section II-A, generally there is more than one ranked list with the same accuracy or AUC value. Thus there are more than one profit value for a set of examples with a fixed accuracy or AUC value. We will use the mean profit for fixed AUC values and the mean profit for fixed accuracy values in our comparison.

As AUC and accuracy are two different measures, we also need to choose an appropriate "mapping" between AUC and accuracy values. Since for a fixed accuracy there are many different AUC values corresponding to it, associating the accuracy value with the average of all AUC values that correspond to this accuracy seems to be a reasonable mapping. For example, if the corresponding average AUC value is 60% for an accuracy at 70%, then we will compare the mean profit with accuracy value at 70% to the mean profit with AUC value at 60%.

For the balanced dataset, we can derive from lemma 1 in [22] that for a given accuracy, the average of all corresponding AUC values of this accuracy is equal to that accuracy value. Therefore for balanced datasets we can directly compare the mean profits for the same AUC and accuracy value.

We perform experiments on balanced ranked list with 20 examples (e.g., customers). We compare the mean profit values for the same AUC and accuracy values, and we only consider the case that AUC and accuracy are equal or greater than 50%. We generate randomly one million ranked lists under the uniform distribution. Our calculation of AUC and accuracy and profits are all based on these ranked lists. We run this experiment on two different promotion cut-off values, promoting only top 25% and top 15% of the top ranked customers.

Figure 2 illustrates the experimental results. For the balanced ranked list of 20 customers, there are 100 AUC values and 10 accuracy values in total. We plot the mean profits for each available AUC values at 51%, 52%, ..., and 100%, and the mean profits for each available accuracy values at 50%, 60%, 70%, 80%, 90%, and 100%.

From Figure 2, we can see that for both promotion cut-off values 25% and 15%, the corresponding mean profits with AUC and accuracy are same when AUC and accuracy are 50%. However, when AUC and accuracy are greater than 50%, the mean profit of AUC *is always greater* than that of accuracy. With the increase of AUC and accuracy, the difference of the

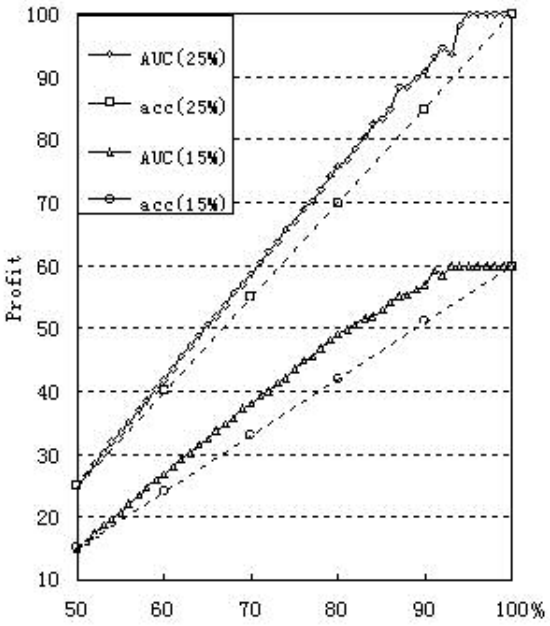

Fig. 2. The mean net profit for AUC and accuracy in a direct marketing study

profit values also increases.

For the promotion cut-off of 15%, the mean profit difference between the AUC and accuracy values at 60%, 70%, 80%, and 90% are 2.7, 5.0, 7.0, and 4.8 respectively. For the promotion cut-off of 25%, the mean profit difference at 60%, 70%, 80%, and 90% are 1.9, 3.8, 5.7, and 6.0 respectively. Therefore except for the values at 90%, the mean profit difference between AUC and accuracy for the 15% promotion cut-off is greater than the mean profits difference between AUC and accuracy for the 25% promotion cut-off. In general the larger the promotion cut-off, the smaller the difference in the mean profit between AUC and accuracy. If the promotion cut-off is 50%, we would not expect to see profit difference, as optimizing accuracy is the same as optimizing the profit.

We can also see from Figure 2 that the slope of the profit curve of AUC is steeper (unless at the maximum profit) than the slope of the profit curve of accuracy. This suggests that improving AUC in a learning algorithm brings larger mean profit than improving accuracy by the same amount. This suggests that learning algorithms optimizing AUC should bring more net benefits to the company than those optimizing accuracy. This reinforces further our conclusion that that AUC is better than accuracy in comparing, choosing, and optimizing different classification

algorithms in real-world applications.

Last, we can also find that for the 15% promotion cut-off, the mean profit of AUC reach the maximum profit when AUC is greater than or equal with 93%, while this is reached only when accuracy is 100%. This shows again that AUC is a better measure to optimize for learning algorithms to reach the highest net profit than accuracy.

From these experiments, we can draw the following conclusions.

- To promote top X% (X $<$ 50) of customers in balanced datasets, the mean profit of a specific accuracy value is less than the mean profit of the average AUCs corresponding to the accuracy. Therefore we can generally say that AUC is a better measure than accuracy when it is used optimizing the net benefit in direct marketing campaigns.
- The difference in the slopes of the AUC and accuracy curves suggests that improving AUC in a learning algorithm brings a larger mean profit than improving accuracy by the same amount.
- With the increase of accuracy or AUC, the mean profit difference also increase. Therefore the advantage of using AUC over accuracy in the promotion profit is more significant when the AUC or accuracy value is higher.
- The difference in the mean profit between accuracy and AUC for a small promotion cut-off is greater than that of large promotion cut-off. Therefore the advantage of using AUC over accuracy in the promotion profit is more significant when the promotion cut-off is small.

To conclude, we have shown through empirical experiments that AUC associates more directly with the net profit of direct marketing campaigns than accuracy. Thus AUC should be favoured over accuracy in real-world applications. Our results may provide important new direction in data mining and its applications.

VI. Conclusions

In this paper, we give formal definitions of discriminancy and consistency in comparing evaluation measures for learning algorithms. We establish precise criteria for comparing two measures in general, and show, both empirically and formally, that AUC is a better measure than accuracy. We then reevaluate commonly accepted claims in machine learning based on accuracy using AUC, and obtain interesting and surprising new results. Last, we show that AUC is more directly associated with the net profit than accuracy in direct marketing. This suggests

that optimizing AUC is preferred over optimizing accuracy in applying machine learning and data mining algorithms to real-world applications.

The conclusions drawn in this paper can have important implications in evaluating, comparing, and designing learning algorithms. In our future work, we will redesign accuracy-based learning algorithms to optimize AUC. Some work has already been done in this direction.

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. Jianning Wang, Dansi Qian, and Huajie Zhang also helped us at various stages of the experiments.

REFERENCES

[1] C. Blake and C. Merz. UCI repository of machine learning databases. http://www.ics.uci.edu/∼mlearn/MLRepository.html, 1998. University of California, Irvine, Dept. of Information and Computer Sciences.

[2] B. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Conference on Computational Learning Theory*, pages 144–152, 1992.

[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] M. Brown, W. Grundy, D. Lin, and N. C. et al. Knowledge-based analysis of microarray gene expression data using support vector machines. In *Proceedings of the National Academy of Sciences*, pages 262–267, 2000.

[5] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

[6] C. C. Chang and C. Lin. Libsvm: A library for support vector machines (version 2.4), 2003.

[7] W. W. Cohen, R. E. Schapire, and Y. Singer. Learning to order things. *Journal of Artificial Intelligence Research*, 10:243–270, 1999.

[8] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[9] P. Domingos and M. Pazzani. Beyond independence: conditions for the optimality of the simple Bayesian classifier. In *Proceedings of the Thirteenth International Conference on Machine Learning* , pages 105 – 112, 1996.

[10] U. Fayyad and K. Irani. Multi-interval discretization of continuous-valued attributes for classification learning. In *Proceedings of Thirteenth International Joint Conference on Artificial Intelligence*, pages 1022–1027. Morgan Kaufmann, 1993.

[11] C. Ferri, P. A. Flach, and J. Hernandez-Orallo. Learning decision trees using the area under the ROC curve. In *Proceedings of the Nineteenth International Conference on Machine Learning (ICML 2002)*, pages 139–146, 2002.

[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*, 45:171–186, 2001.

[13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[14] C. Hsu and C. Lin. A comparison on methods for multi-class support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2001.

[15] J. Huang, J. Lu, and C. X. Ling. Comparing naive bayes, decision trees, and svm using accuracy and auc. In *Proceedings of the 3rd International Conference on Data Mining(ICDM-2003)*, page To appear, 2003.

[16] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of European Conference on Machine Learning*, pages 137–142, 1998.

[17] I. Kononenko. Comparison of inductive and naive Bayesian learning approaches to automatic knowledge acquisition. In B. Wielinga, editor, *Current Trends in Knowledge Acquisition*. IOS Press, 1990.

[18] P. Langley, W. Iba, and K. Thomas. An analysis of Bayesian classifiers. In *Proceedings of the Tenth National Conference of Artificial Intelligence*, pages 223–228. AAAI Press, 1992.

[19] Y. Lin. Support vector machines and the bayes rule in classification. *Data Mining and Knowledge Discovery*, 6(3):259–275, 2002.

[20] C. Ling and C. Li. Data mining for direct marketing - specific problems and solutions. In *Proceedings of Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 73–79, 1998.

[21] C. Ling and J. Yan. Decision tree with better ranking. In *Proceedings of 2003 International Conference on Machine Learning (ICML'2003)*, 2003.

[22] C. X. Ling, J. Huang, and H. Zhang. AUC: a statistically consistent and more discriminating measure than accuracy. In *Proceedings of 18th International Conference on Artificial Intelligence (IJCAI-2003)*, pages 329–341, 2003.

[23] C. X. Ling and H. Zhang. Toward Bayesian classifiers with accurate probabilities. In *Proceedings of the Sixth Pacific-Asia Conference on KDD*, pages 123–134. Springer, 2002.

[24] H. Liu, F. Hussain, C. L. Tan, and M. Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 6(4):393–423, 2002.

[25] D. Meyer, F. Leisch, and K. Hornik. Benchmarking support vector machines. Technical report, Vienna University of Economics and Business Administration, 2002.

[26] F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*, 2003. To appear.

[27] F. Provost and T. Fawcett. 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*, pages 43–48. AAAI Press, 1997.

[28] 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. Morgan Kaufmann, 1998.

[29] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann: San Mateo, CA, 1993.

[30] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[31] P. Smyth, A. Gray, and U. Fayyad. Retrofitting decision tree classifiers using kernel density estimation. In *Proceedings of the 12th International Conference on machine Learning*, pages 506–514, 1995.

[32] J. A. K. Suykens and J. Vandewalle. Multiclass least squares support vector machines. In *IJCNN'99 International Joint Conference on Neural Networks*, Washington, DC, 1999.

[33] V. Vapnik. *Statistical Learning Theory*. Wiley, NY, 1998.