PERFORMANCE MEASURES OF MACHINE LEARNING

(Spine title: Performance Measures of Machine Learning)

(Thesis format: Monograph)

by

Jin Huang

Graduate Program
in
Computer Science

A thesis submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy

Faculty of Graduate Studies
The University of Western Ontario
London, Ontario, Canada

Your file Votre référence
ISBN: 978-0-494-30363-4
Our file Notre référence
ISBN: 978-0-494-30363-4

THE UNIVERSITY OF WESTERN ONTARIO
FACULTY OF GRADUATE STUDIES

CERTIFICATE OF EXAMINATION

Supervisor	Examiners
Dr. Charles Ling	Dr. Sylvia Orborn
Supervisory Committee	Dr. Kamran Sedig
	Dr. Martin Houde
	Dr. Aijun An

The thesis by

Jin Huang

entitled:

Performance Measures of Machine Learning

is accepted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Date_____

_____
Chair of the Thesis Examination Board

Abstract

This thesis investigates some fundamental issues of performance measures of machine learning .

Performance measures (or evaluation measures) play important roles in machine learning. They are not only used as the the criteria to evaluate learning algorithms, but also used as the heuristics to construct learning models. However, little work has been done to thoroughly explore the characteristics of performance measures.

We first formally propose criteria to compare performance measures. These criteria focus on studying the consistency relationship between two measures, and whether one measure has more discriminatory power than the other. Based on the proposed criteria, we theoretically and empirically compare two most popular measures: accuracy and AUC (Area Under the ROC Curve). We show that AUC is statistically consistent and more discriminant than accuracy, which indicates that AUC should be preferred over accuracy in evaluating learning algorithms. We also compare ranking measures and give a preference order to use these measures in comparing ranking performance.

Based on the comparison criteria, we propose two general approaches to construct new measures from existing measures. We formally prove that the new measures are consistent and more discriminant than the existing ones. We also compare the learning models of artificial neural networks trained with the newly constructed measures and existing measures. The experiments show that the model trained with the newly constructed measure outperforms the models trained with the existing measures.

Finally, we explore model selection tasks using measures. We show that generally we should use different measures as model selection goal and evaluation measures. We show that a measure's model selection ability is stable to model selection goal and class distributions. We find that some measures perform better than others in the

model selection tasks.

In summary, this thesis addresses several fundamental issues of machine learning measures. The research results are very useful in real world applications. It provides the guidance on how to select suitable measures to evaluate learning algorithms. Furthermore, it also presents general approaches to construct new measures efficiently and effectively, which provides new approaches in building learning models.

Keywords: performance measures, threshold measures, ranking measures, probability-based measures, comparison criteria, constructing measures, model selection

Acknowledgements

I would like to thank Dr. Charles Ling, my supervisor, who had the greatest role in my thesis research in the past four years. I will always be grateful for all his guidance and support for my research in the interesting topics of this thesis. Whithout his instruction, it's hard to imagine that I can finish this thesis.

I thank the examiners of my thesis proposal, Dr. Bob Mercer and Dr. Sylvia Osborn. They carefully read my proposal and gave me valuable suggestions to improve my work.

Thanks are also given to members of our Data Mining Lab at Western: Jun Yan and Shengli Sheng. They are great collegues and provided me with a friendly environmemnt to work in.

Finally, I would like to give my deepest thanks to my wife Jing. She gave me unconditional encouragement and support in helping me finish my thesis.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many performance measures are widely used in the fields of machine learning, knowledge discovery and data mining. They are primarily used for two purposes. First, they are used as criteria to compare and evaluate machine learning algorithms. Traditionally, a variety of measures, such as accuracy, precision, recall, F-measure, and so on are adopted as criteria to evaluate the performance of information systems. For example, in classification tasks accuracy is defined as the percentage of objects that are correctly classified. It measures the classification performance of a learning algorithm. Most previous research used accuracy as the evaluation criterion. In the last two decades, accuracy was the most frequently used measure in algorithm performance evaluation. As another example, in information retrieval, precision and recall are two traditional measures that are used to evaluate the query quality. In recent years, AUC (See Section 2.2.1) is becoming another popular measure in machine learning.

Second, performance measures are also used as heuristics to construct or optimize learning algorithms. For example, Information Gain is used as the heuristic to construct decision trees. Least Squared Error is the heuristic for training neural networks. In recent years, some researchers also used AUC to optimize decision trees [22], artificial neural networks [65], and support vector machines [52].

However, although performance measures are very important in machine learning, little work has been done to thoroughly and systematically explore the characteristics of measures. Choosing measures for a certain context is largely determined historically. It is still not very clear why one measure performs well in a specific situation, while

it has poor performance in another condition. In this thesis, a significant framework that aims at comparing and evaluating machine learning measures is proposed. Our work is primarily focused on several fundamental issues of machine learning measures: comparing measures, constructing better measures, and model selection with measures.

We can make a crude analogy of machine learning measures with measures used to evaluate university students. Many different measures, such as numerical marks, letter marks, and pass or fail, can be used to evaluate students' performance. Are they consistent? Which measure is better than others, and why? This is analogous to the comparison of machine learning measures studied in the thesis. What are the learning strategies of students in order to optimize certain measures? This is analogous to using better measures for constructing better learning algorithms. What measures would we use to select the best students according to a certain measure? This is analogous to model selection using machine learning measures.

The rest of this chapter is organized as follows. We first review some commonly used learning measures (Section 1.1). We then introduce our work of comparing measures (Section 1.2). We briefly describe the ideas of how to construct better measures (Section 1.3). We also introduce our work of evaluating model selection abilities of measures (Section 1.4). Finally, we list our major contributions in this thesis (Section 1.5).

1.1 Machine Learning Measures

To give a detailed description of different measures, we first introduce some concepts used to compute performance measures. A measure is calculated from the given dataset or the classification results produced by a classifier. In a classification task, many learning algorithms, such as decision trees, Naive Bayes, not only predict labels, but also produce the probabilities of belonging to different classes for all examples. Usually the predicted results combined with a predefined threshold can be used to compute the performance measures.

Suppose a binary (with only positive and negative classes) dataset with P positive and N negative examples is classified. Each example is predicted with a probability of belonging to the positive class. We rank the classified examples according to their

predicted probabilities of belonging to the positive class and we set a probability threshold. Any example whose predicted probability being positive is above the threshold is regarded as a positive example. Therefore if an example's true label is positive, it is called a true positive example; otherwise it is called a false positive example. Similarly, we have the meaning of true negative example and false negative example.

The ratio of the true positive examples' number to total positive examples number (P) is called the true positive rate, which is represented with TP. The ratio of the true negative examples number to total negative examples number (N) is called the true negative rate, which is represented with TN. The false positive rate FP and false negative rate FN can also be similarly defined.

According to the concepts we have introduced, we can conveniently define different measures. The measures can be categorized into three basic types: Threshold Measures, Ranking Measures, and Probability Based Measures.

1.1.1 Threshold Measures

This type of measures share the same point that they are computed based on predefined thresholds. Thus they are called threshold measures. For binary classification usually a default probability threshold of 0.5 is adopted. For example, Table 1.1 shows a binary ranked list with 4 positive (+) and 6 negative (−) examples. The threshold is set in the middle of this ranked list.

Many commonly used measures can be categorized into this type, including accuracy, precision, recall and F-score, and so on.

Table 1.1: A binary ranked list with 4 positive and 6 negative examples.

−	−	−	−	+	|	−	+	+	−	+

Accuracy: This measure is the most widely used performance measure in Machine Learning. For a given dataset, it is defined as the proportion of correct predictions to the size of the dataset.

$$accuracy = \frac{P * TP + N * TN}{P + N}$$

In Table 1.1 we can calculate $accuracy = \frac{3+4}{4+6} = \frac{7}{10}$.

Precision: This measure comes from Information Retrieval. It measures the proportion of predicted positive examples that are actually positive.

$$precision = \frac{P * TP}{P * TP + N * FN}$$

In Table 1.1 we can calculate $precision = \frac{3}{5}$.

Recall: This measure also comes from Information Retrieval. It measures the proportion of positive examples that are actually predicted as positive, which is exactly the TP.

$$recall = TP$$

In Table 1.1 we can calculate $recall = \frac{3}{4}$.

F-score: Usually precision and recall are used to evaluate performance simultaneously. To use a single measure for evaluation, F-score is introduced to combine precision and recall. It is defined as the harmonic mean of the precision and recall.

$$F = \frac{2 * precision * recall}{precision + recall}$$

In Table 1.1 we can calculate $F = \frac{2\frac{3}{5}\cdot\frac{3}{4}}{\frac{3}{5}+\frac{3}{4}} = \frac{2}{3}$.

Lift: Similar to precision, lift also measures the proportion of predicted positive examples that are actually positive. The difference with precision is that usually its threshold is set at the position that a fixed percentage of the dataset is classified as positive.

$$lift = \frac{P * TP}{P * TP + N * FN}$$

In Table 1.1 we can calculate $lift = \frac{3}{5}$.

Break-even point(BEP): This measure is defined as the precision value at the threshold that precision equals recall. In Table 1.1, when the threshold is set in the position between the 6th and 7th examples (the examples are numbered from left to right), precision equals recall. In this case, $BEP = precision = recall = \frac{3}{4}$.

In Chapter 3 we will compare the measures of accuracy and AUC (See Section 2.2.1). In Chapter 5 we will study the model selection abilities of the measures of F-score, Lift, and Break-even point.

Table 1.2: A predicted binary ranked list

	−	−	−	−	+	−	+	+	+	+
i					1		2	3	4	5
r_i					5		7	8	9	10
p_i^+					0.60		0.73	0.81	0.88	0.90
p_j^-	0.20	0.39	0.44	0.57		0.65				

1.1.2 Ranking Measures

This type of measures have the common property that they are evaluated based on the relative ordering relations of the examples. In many machine learning and data mining applications, ranking is more desirable than simple classification. Thus it is quite important to choose suitable ranking measures to evaluate the ranking performance of different algorithms. Generally, there are two types of ranking measures.

The first type of ranking measures are the "partial" or "censored" ranking measures. These measures only consider the relative ordering relations of examples that belong to different classes. The Area Under the ROC Curve (AUC) can be categorized into this type. The average precision (APR) is also categorized in this type of ranking measures because it is closely correlated with AUC. We also propose a new measures, SAUC, which is a variant measure of AUC.

The first row in Table 1.2 shows a binary predicted ranked list with 5 positive and 5 negative examples, and some notations used to define AUC and SAUC. In this ranked list, only the relative orderings of examples belonging to different classes are considered. In rest rows of Table 1.2 we use i to represent the ith positive example, r_i to represent the ranked position of the ith positive example. We also use p_i^+, p_j^- to represent the predicted probabilities of being positive for the ith positive example and the jth negative example, respectively.

For the following definitions, we assume that there are n_0 positive and n_1 negative examples in the binary ranked list.

AUC: The Area Under the ROC Curve, or simply AUC, is a single-number measure widely used in evaluating classification algorithms. Researchers have found that AUC exactly reflects the overall ranking performance of a classifier. For a binary ranked list, Hand and Till [25] present the following simple approach to calculating AUC

$$AUC = \frac{S_0 - n_0(n_0+1)/2}{n_0 n_1}, \tag{1.1}$$

where $S_0 = \sum r_i$. The AUC of the ranked list in Table 1.2 is $\frac{(5+7+8+9+10)-5\times6/2}{5\times5}$, which is 24/25.

Clearly formula 1.1 can be rewritten as

$$AUC = \frac{\sum_{i=1}^{n_0}(r_i - i)}{n_0 n_1} \tag{1.2}$$

$(r_i - i)$ can be viewed as the number of negative examples ranked behind the ith positive example. If we define

$$I(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$$

then the formula of AUC can be written as

$$AUC = \frac{\sum_{i=1}^{n_0}\sum_{j=1}^{n_1} I(p_i^+ - p_j^-)}{n_0 n_1} \tag{1.3}$$

This shows that AUC reflects whether each positive example is ranked higher or lower than each negative example. We will give detailed arguments about how the statistical meaning and calculation formula of AUC are obtained in Section 2.2.1 of Chapter 2.

SAUC: We propose a new measure, SAUC (Softened Area Under the ROC Curve). SAUC is defined as

$$SAUC = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n}(p_i^+ - p_j^-)I(p_i^+ - p_j^-)}{n_0 n_1} \tag{1.4}$$

In Table 1.2 we can easily calculate $SAUC = \frac{5.64}{25} = 0.2256$.

Clearly, SAUC is in the range of [0,1]. The closer the predicted probabilities to the true probabilities, the larger the SAUC. SAUC and AUC have the common point in that they both measure how each positive instance is ranked compared with each negative instance. However, AUC only cares whether each positive instance

is ranked higher or lower than each negative instance, while SAUC also considers the probability differences in the ranking. In addition, SAUC also reflects how well the positive instances are separated from the negative instances according to their predicted probabilities. Thus SAUC can be categorized both as a ranking and a probability-based measure. As a more refined and delicate measure than AUC, SAUC can reflect both ranking and probability predictions. In Chapter 5 we will show that SAUC usually can achieve excellent model selection ability.

APR: Strictly speaking, average precision is not a ranking measure. Since research has shown that it is closely correlated to AUC, here we categorize it into the ranking measures.

For a binary ranked list, APR is defined as the average of all precision values when the decision thresholds are set on the ranked positions of different positive examples.

$$APR = \frac{1}{n_0} \sum_{i=1}^{n_0+n_1} \frac{a_i}{n_0 + n_1 - r_i} \tag{1.5}$$

where a_i is the number of positive examples that are ranked higher than the position of r_i. In Table 1.2 the APR can be computed as $\frac{1}{5}(1+1+1+1+\frac{5}{6}) = 0.967$.

In Chapter 5 we will study the model selection ability of APR.

The second type ranking measures is the true ranking measures, which takes into consideration the relative ordering relation between every two examples in a ranked list in evaluating a ranking performance. The commonly used true ranking measures include Euclidean Distance (ED), Manhattan Distance (MD), Sum of Reserved Number (SRN). A new true-ranking measure, OAUC, is also proposed.

For a true ranked list with n examples, the actual ranked positions for all examples are $n, n-1, \cdots, 1$. For the example whose actual ranking position is i, we use a_i to denote its predicted ranking position. Table 1.3 provides an example with a true ranked list and a predicted ranked list.

Euclidean Distance (ED): If we consider the true ranking list and the predicted ranking list as two points of $(1, 2, \ldots, n)$ and $(a_1, a_2, \ldots, a_n)$ in an n-dimensional Euclidean space, then Euclidean Distance between these two points are

Table 1.3: An example of a true ranked list and its predicted ranked list

True	1	2	3	4	5	6	7	8
	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
Predicted	3	6	8	1	4	2	5	7

$$ED = \sqrt{\sum_{i=1}^{n}(a_i - i)^2} \tag{1.6}$$

For the example in Table 1.3, It is easy to obtain that $ED = (3-1)^2 + (6-2)^2 + (8-3)^2 + (1-4)^2 + (4-5)^2 + (2-6)^2 + (5-7)^2 + (7-8)^2 = 76$.

Manhattan Distance (MD): We also consider the true ranking list and the predicted ranking list as two points in an n-dimensional Euclidean. The Manhattan Distance is defined as

$$MD = \sum_{i=1}^{n}|a_i - i| \tag{1.7}$$

For the example in Table 1.3, it is easy to obtain that $MD = |3-1| + |6-2| + |8-3| + |1-4| + |4-5| + |2-6| + |5-7| + |7-8| = 22$.

Sum of Reversed Number (SRN): This is roughly the sum of the reversed pairs in the list. That is,

$$SRN = \sum_{i=1}^{n} s(i) \tag{1.8}$$

For the ith example, its reversed number $s(i)$ is defined as the number of examples whose positions in predicted ranking list are greater than i, but for which the actual ranked positions are less than i. For the example in Table 1.3, we can find that the examples of 1 and 2 are both ranked higher than the first example 3 in predicted list. Thus $s(1) = 1+1 = 2$. Similarly we have $s(2) = 4$, $s(3) = 5$, etc. Therefore the SRN for the ordered list $\pi(l)$ is $SRN = 2+4+5+0+1+0+0+0 = 12$.

Ordered AUC (OAUC): We propose a new measure called Ordered Area Under Curve (OAUC), as it is similar to AUC both in meaning and calculation. The dif-

ference is that OAUC is a true-ranking measure, while AUC is a partial ranking measure. The formula of OAUC is similar with AUC, except that each term in the OAUC formula is weighted by its true order, and the sum is then normalized.

OAUC is defined as follows:

$$OAUC = \frac{\sum a_{r_i}(r_i - i)}{\lfloor \frac{n}{2} \rfloor \sum_{i=1}^{\lceil \frac{n}{2} \rceil} (\lfloor \frac{n}{2} \rfloor + i)} \tag{1.9}$$

.

In the ranked list in Table 1.3, the positive examples are 5, 6, 7, 8 which are positioned at 7, 2, 8 and 3 respectively. Thus $r_1 = 2$, $r_2 = 3$, $r_3 = 7$, $r_4 = 8$, and $a_{r_1} = 6$, $a_{r_2} = 8$, $a_{r_3} = 5$,$a_{r_4} = 7$.

$$OAUC = \frac{6(2-1) + 8(3-2) + 5(7-3) + 7(8-4)}{4((4+1) + (4+2) + (4+3) + (4+4))} = \frac{31}{52}$$

.

It is interesting to discuss the similarity and difference between OAUC with AUC, SAUC. Clearly, from formula 1.9 we can see that OAUC uses larger weights on the higher ranked examples, while AUC uses equal weights on all examples. This indicates that OAUC is biased to reflect the ranking performance of some highest ranked examples. AUC reflects the overall ranking performance for all ranked examples. SAUC can be viewed as the probability version of AUC, since it incorporates the predicted probabilities in calculating AUC.

1.1.3 Probability-based Measures

Apart from the above two kinds of measures, some measures are based on probability estimations. Many traditional measures, such as RMS (Root Mean Squared Error), Mean Cross Entropy (MXE), and Information Gain (IG) belong to this category.

RMS: RMS is widely used in regression. It measures the amount of predictions that deviate from true targets. For K instances, suppose that the true probability value and the predicted probability value for an instance I_i are $Tar(I_i)$ and $Pred(I_i)$,

$$RMS = \sqrt{\frac{1}{K}\sum_{i=1}^{K}[Tar(I_i) - Pred(I_i)]^2}$$

MXE: MXE is used in the probabilistic setting when interested in predicting the probability that an instance is positive. It can be proved that in this setting minimizing the cross entropy gives the maximum likelihood hypothesis.

$$MXE = -\frac{1}{K}\sum_{i=1}^{K}\{Tar(I_i) * log[Pred(I_i)] + (1 - Tar(I_i)) * log[1 - Pred(I_i)]\}$$

In Chapter 5 we will study the model selection abilities of the measures of RMS and MXE.

1.2 Comparing Machine Learning Measures

Different measures are widely used in different machine learning, data mining algorithms and applications. For example, entropy (a kind of error-based measure) measures are traditionally used as decision tree splitting criteria. Precision, recall, and F-measure are widely adopted for evaluating the performance of text mining. However, it is still not very clear why we choose a measure for a specific application. There is little knowledge about under which conditions one measure performs better than others.

One approach to answer these questions is to study the relations among different measures. Some research has been done in this direction. Caruana and Niculescu-Mizil [9] empirically compared the correlations among some widely used measures. They concluded that RMS is mostly correlated with other measures on average and thus is the most reliable measure when the best measure is unknown.

In this thesis, we propose a framework to compare measures. We compare two arbitrary measures in two different aspects. First, we study the consistency between two measures. That is, when using two measures f and g to evaluate objects a and b, if f says that a is better than b and g also says that, we claim that f and g are consistent. Otherwise they are inconsistent. Usually two measures are consistent in evaluating some objects but inconsistent in evaluating other objects. In this case we investigate the statistical consistency, which is the portion of consistent objects pairs to all objects pairs. Second, we compare the discriminatory power between two arbitrary measures. The discriminatory power of one measure shows how well it can

discriminate among different objects. For example, when evaluating university student scores we usually adopt two scoring systems: numerical marks and letter marks. The numerical marks have scores of 0, 1, 2, $\cdots$, 100 while the letter marks have scores of A, B, C, D, F. Clearly letter marks and numerical marks are consistent, and numerical marks have much more discriminatory power than that of letter marks because numerical marks can reach much more different scores.

To address the above two issues we formally propose several criteria to compare the consistency and discriminancy between two arbitrary measures. These criteria give detailed and complete comparison of the consistency, inconsistency, discriminancy and indifferency among arbitrary measures. We give detailed arguments in Chapter 3.

Based on these criteria, we first theoretically and empirically compare the measures of accuracy and AUC. We then compare some popular ranking measures.

1.2.1 Comparing AUC and accuracy

Accuracy is the most widely used machine learning measure. In measuring the quality of a classification task, it reflects how many data instances are correctly classified. Traditionally the performance of most learning algorithms is evaluated by accuracy. In some algorithms it is used as a heuristic for model optimization or construction. For example, accuracy is one of the splitting criteria for building decision trees. Accuracy is also the heuristic for building linear classification models.

However, in some applications accuracy is not enough. For example, in data mining applications, direct marketing desires a ranking of the customers according to their likelihood of purchasing. In this case we need to measure how well the customers are ranked. A new measure that reflects the ranking performance is required.

The ROC (Receiver Operating Characteristics) Curve was originally used in signal processing to depict the trade-offs between the hit rates and alarm rates [24, 19]. In recent years, it was introduced into the machine learning community. The AUC (Area Under the ROC Curve) is a one-number measure which reflects the general ranking performance of a learning algorithm. This number is widely used in various engineering, scientific and medical applications. Recently in the machine learning and data mining communities it has gained an increasing acceptance in comparing learning algorithms [48] and constructing learning models [22, 39].

However, accuracy is traditionally designed to judge the merits of classification results, and AUC is simply used as a replacement of accuracy without much reasoning for why it is a better measure, especially for the case of ordering. The main reason for this lack of understanding is that up to now, there has been no theoretical study on whether any of these measures work better than others.

In this thesis, we will use a set of comparison criteria to give theoretical and empirical comparisons between accuracy and AUC. We formally prove that AUC is statistically consistent and more discriminant than accuracy. Empirically, we perform experiments by using artificial datasets, real-world datasets, and some popular machine learning algorithms to confirm our theoretical results. Finally, we reevaluate some popular machine learning algorithms by using AUC instead of accuracy.

1.2.2 Comparing Ranking Measures

Ranking of cases is an increasingly important way to describe the results of many data mining and other science and engineering applications. For example, the results of document searches in information retrieval and Internet search is typically a ranking of the results in the order of match. Customer relationship management (CRM) applications typically rank the customers in order of desirability. This leaves two issues to be addressed. First, given two orders of cases, how do we design or choose a measure to determine which order is better? Second, given two different ranking measures, how do we tell which measure is more desirable?

As we have discussed in section 1.1, there are generally two types of ranking measures: true ranking measures and "partial" ranking measures. In this thesis, we use the consistency and discriminancy criteria to compare these two kinds of ranking measures. We perform two experiments to explore their consistency and discriminancy relations. We first generate artificial ranked lists with various lengths to empirically compute the statistical consistency and discriminancy among measures. We then use real-world datasets and two learning algorithms to study the discriminatory powers of ranking measures. From these experiments we can obtain a preference order of the ranking measures. We conclude that OAUC and ED are the ranking measures with the highest discriminatory power. The MD has the weakest discriminatory ability.

1.3 Constructing Better Measures for Machine Learning

New measure designing is another important issue in machine learning. Historically the designing of measures highly depends on specific applications or domains. There currently exists little work proposing general approaches to build new measures.

In this thesis, based on our consistency and discriminancy criteria, we propose two novel methods to construct new measures. These two approaches are linear combination and two-level construction. The former uses the weighted linear combination of two measures to construct a new measure. The latter constructs a two-level measure. We formally prove that the new measures constructed with either approach are consistent and more discriminant than the existing ones.

Following these approaches we construct two new measures by using AUC and accuracy as the base measures. We empirically show that the new measures are more closely correlated with the external measures RMS than AUC and accuracy. Research has shown that RMS is a robust and well performed measure. Thus the new measures are expected to perform better than AUC and accuracy.

We then use accuracy, AUC, and the two-level measure constructed from AUC and accuracy as heuristics to train artificial neural network models from real-world datasets, respectively. The experimental results show that the models trained with the two-level measure perform better than the models trained with AUC, and significantly better than the model trained with accuracy. This confirms the advantages of the new measures. It provides a new approach to build better learning models.

1.4 Model Selection with Measures

Model selection is an important task in machine learning. The goal of model selection is to select the model with the best expected performance among a given set of models. A consensus in the machine learning community is that the same model selection goal should be used to identify the best model based on available data. However, following the preliminary work of Rosset [53], we show that this is, in general, not true under highly uncertain situations where only very limited data are available. We thoroughly investigate model selection abilities of different measures under highly

uncertain situations as we vary model selection goals, learning algorithms and class distributions. The experimental results show that a measure's model selection ability is relatively stable to the model selection goals and class distributions. However, different learning algorithms call for different measures for model selection. For learning algorithms of Support Vector Machines and K-nearest neighbor, generally the measures of RMS, SAUC, MXE perform the best. For learning algorithms of decision trees and Naive Bayes, generally the measures of RMS, SAUC, MXE, AUC, APR have the best performance.

1.5 Contributions of the Thesis

Measures are quite important in machine learning and data mining. However there is almost no systematic study on the relations of different measures. In this thesis, we propose a framework to do some fundamental work.

In Chapter 3, we first formally propose a set of criteria with the goal of comparing the performance of two arbitrary measures. These criteria are focused on two aspects of a measure's characteristic: consistency and discriminancy. Consistency depicts whether or not two measures are consistent in evaluating objects. Discriminancy reveals how well one measure can discriminate different objects. Based on these criteria we compare two frequently used machine learning measures: AUC and accuracy. We show theoretically that AUC is consistent and more discriminant than accuracy. This work was published in the *Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03)* [37].

The theoretical study is only for the case of binary and balanced datasets. To see whether this theoretical result is also true for the case of imbalanced and multi-class datasets, we perform experiments with imbalanced, multi-class artificial datasets. The empirical results show that our theoretical results can be extended to more general cases. This work was published in the *Proceedings of the Third International Conference on Data Mining (ICDM-03)* [32] and *Proceedings of 2003 Canadian Artificial Intelligence Conference* [38].

We also perform an empirical study by using real world datasets and some popular machine learning algorithms. The experiments on real world datasets also confirm our theoretical results. We then use AUC and accuracy to evaluate the popular learning algorithms of decision trees, Support Vector Machines and Naive Bayes. Previous

researches showed that these algorithms perform quite similarly when evaluating with accuracy. However, we show that they perform significantly differently when evaluating with AUC. The combination of the above work was published in *IEEE transactions on Knowledge and Data Engineering, V.17 No.3 pp.299-310, March 2005* [29].

We next compare ranking measures. Some commonly used ranking measures have been studied individually in Statistics. However, it is still not clear how they are correlated with each other. We use the criteria proposed to study their consistency and discriminancy relations. We obtain a preference order of those ranking measures. This work was published in *Proceedings of the 9th European Conference on Practice and Principle of Knowledge Discovery in Database (PKDD-05)* [30].

The second issue of this thesis is new measure design. We address this issue in Chapter 4. Most previous work in this issue highly depends on specific applications and domains. Here based on our comparison criteria we propose two general approaches to construct new measures based on existing ones. We formally prove that the new measures are more discriminant than the existing ones. We then use the two-level measure formed by AUC and accuracy to train a learning algorithm and compare the performance of the learned model with other traditionally trained model. The result shows that the model trained by two-level measure is significantly better than the traditional model.

The third issue of the thesis is model selection with measures. We address this issue in Chapter 5. Here we study the model selection abilities of measures when only limited data are available. We show that generally the model selection goal measure should not be used to evaluate different models. We also show that a measure's model selection ability is relatively stable to model selection goals and class distributions. However, different learning algorithms call for different measures. This work will be published in *the Workshop on Evaluation Methods for Machine Learning at the 21st National Conference on Artificial Intelligence (AAAI-06)* [31]. It was also submitted to *the 17th European Conference on Machine Learning (ECML-06).*

Chapter 2

Review of Previous Work

Compared with other widely studied topics in machine learning and data mining, only a few studies investigated the relationships among various machine learning measures. We first give a brief review of some popular learning algorithms such as decision trees (C4.5, C4.4), Naive Bayes, Support Vector Machines and Artificial Neural Networks as they will be used in the experiments of the following chapters. We then review the previous work on measures.

2.1 Machine Learning Algorithms

2.1.1 Decision Trees

Decision tree learning is one of the most widely used classification algorithms. A decision tree is a top-down tree with a recursive structure. There are two types of nodes in a decision tree: leaf nodes and internal nodes. Leaf nodes represent the classification labels. Internal nodes represent the partitions of all the examples according to some attribute values. A decision tree classifies examples by sorting them down from root to leaf nodes. Starting from the top of the tree, an incoming example is tested by the attribute specified by the root node, then tested down to one of its children nodes corresponding to the attribute values of the incoming example. This process is repeated until a leaf node is reached and this example is classified as the class represented by the leaf node. The most successful decision tree algorithms

are ID3 and its successor C4.5, which were developed by Quilan [51]. Provost and Domingos [48] proposed an improved version of C4.5, which is called C4.4.

2.1.1.1 C4.5 Algorithm

Decision tree building is a recursive process that applies a greedy search through the space of possible attributes. Let S be the collection containing training examples. The k possible values of the class attribute are denoted as $C_1, C_2, \cdots, C_k$. The C4.5 algorithm can be described as follows:

- Create a Root node for the tree using all the examples in S.
- If all examples in S belong to a single class value C_j, return the single-node tree Root, which is labeled as C_j.
- If there are no examples in S, return the single-node tree Root, which is labeled as most frequent class value at its parent node.
- If examples in S belong to more than one class value, select an attribute A that best classifies the examples, Assign attribute A for Root. For each value V_i of A, create a new branch below Root. Create a new subset S_i of S that has V_i for A. Apply the same procedure to S_i recursively.

At each step, the decision tree building algorithm tries to find the best attribute that splits the training set into several subsets which have the same attribute value, then applies the same to each subset recursively until the tree classifies the examples perfectly.

This algorithm grows each branch of the tree just deeply enough to perfectly classify the training examples. However this strategy sometimes will overfit the training examples when there is noise in the data, or when the number of training examples is too small. To overcome this difficulty, C4.5 adopts a tree pruning strategy. That is, pruning the nodes of a large tree to make it more robust to noise or more accurate for unseen testing examples. More specifically, it uses a post-pruning rule as follows:

- Build a decision tree following the above algorithm described.
- Convert the tree into an equivalent set of rules.

- Prune each rule by removing any preconditions that result in improving its estimated accuracy.
- Sort the pruned rules according to estimated accuracies.

Although practically this pruning strategy can prevent overfitting and improve prediction accuracy, it gives poor probability estimation for the training examples. An improvement algorithm was proposed, which is called C4.4.

2.1.1.2 C4.4 Algorithm

Decision trees that produce probability estimations are called PETs (Probability Estimation Trees) [47]. The leaf nodes of a decision tree may contain training examples of different classes. The probability of a testing instance belonging to a specific class is normally the ratio of training instances of that class over all examples in the leaf node that the testing instance falls in.

The popular decision tree learning algorithm C4.5 has been observed to produce poor probability estimations on AUC [58, 50, 48]. Provost and Domingos [48] proposed an improved version, which is called C4.4. They made 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 showed that pruning also reduces the quality of the probability estimation, as discussed above. For this reason, they chose to build the trees 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. We assume that there are a total of N examples in a leaf, in which k examples belong to the positive class. Suppose there are totally C classes. Then the

estimated probability of being positive for this leaf is $\frac{k}{N}$. The Laplace correction calculates this estimated probability as $\frac{k+1}{N+C}$.

They called the resulting algorithm C4.4, and showed that C4.4 produces decision trees with significantly higher AUC than C4.5 [48].

2.1.2 Naive Bayesian Networks

Bayesian network is another traditional learning algorithm. Bayesian networks are probabilistic models that combine probability theory and graph theory [46, 33]. A Bayesian network consists of a structural model and a set of conditional probabilities. The structural model is a directed graph in which nodes represent random variables and arcs represent informational or causal dependencies among the variables. The dependencies are quantified by conditional probabilities for each node given its parents in the network.

Bayesian networks are often used for classification problems. In classification learning problems, a learner attempts to construct a classifier from a given set of training examples with class labels. Assume that A_1, A_2,$\cdots$, A_n are n attributes (attribute nodes in the corresponding Bayesian network) . An example E is represented by a vector $(a_1, a_2, , \cdots, a_n)$, where a_i is the value of A_i.

Let C represent the classification variable (the class node in the corresponding Bayesian network), which takes value $+$ (positive class) or $-$ (negative class). We use c to represent the value that C takes. A classifier is a function that assigns a class label to an example. From the probability perspective, according to Bayes Rule, the probability of an example $E = (a_1, a_2, \cdots, a_n)$ being class c is

$$p(c|E) = \frac{p(E|c)p(c)}{p(E)}.$$

E is classified as the class $C = +$ iff (if and only if)

$$g(E) = \frac{p(C = +|E)}{p(C = -|E)} \geq 1, \tag{2.1}$$

where $g(E)$ is called a Bayesian classifier.

Assume that all attributes are independent given the value of the class variable; that is,

$$p(E|c) = p(a_1, a_2, \cdots, a_n|c) = \prod_{i=1}^{n} p(a_i|c), \tag{2.2}$$

the resulting $g(E)$ is then:

$$g(E) = \frac{p(C=+)}{p(C=-)} \prod_{i=1}^{n} \frac{p(a_i|C=+)}{p(a_i|C=-)}. \tag{2.3}$$

$g(E)$ is called a Naive Bayesian classifier, or simply Naive Bayes (NB).

Because the values of $p(a_i|c)$ can be estimated from the training examples, Naive Bayes is easy to construct. Naive Bayes is the simplest form of Bayesian network, in which all attributes are independent given the value of the class variable. This is called conditional independence. It is obvious that the conditional independence assumption in Naive Bayes is rarely true in many applications. It is, however, surprisingly effective in the classification tasks [35, 36, 45]. Many empirical comparisons between Naive Bayes and modern decision tree algorithms such as C4.5 [51] showed that Naive Bayes predicts equally well as C4.5.

2.1.3 Support Vector Machines

Support Vector Machine (SVM) is a learning algorithm that learns linear functions in the high dimensional feature space. It was first proposed by Vapnik and his team at AT&T Bell Labs [6, 13, 63]. SVM has been shown to be a very powerful method that outperforms most other learning algorithms in a wide variety of applications. Traditional learning algorithms often are trained with the goal of minimizing the training error. SVMs adopt another induction approach, which minimizes the upper bound of the generalization error.

We use a simple binary classification case to introduce the basic idea of SVM learning algorithms. We assume that we have a dataset with labeled examples $\{(x_1, y_1), \cdots, (x_l, y_l)\}$, where $y_i \in \{-1, 1\}$. From the labeled examples we can train a variety of linear classifiers to separate the training examples. We wish to determine the linear classifier with the smallest generalization error. Since each linear classifier corresponds to a hyperplane, a good choice is to find the hyperplane that leaves the maximum margin

between the two classes. The margin is defined as the sum of the distances of the hyperplane from the closest points of the two classes. This is shown in Figure 2.1.

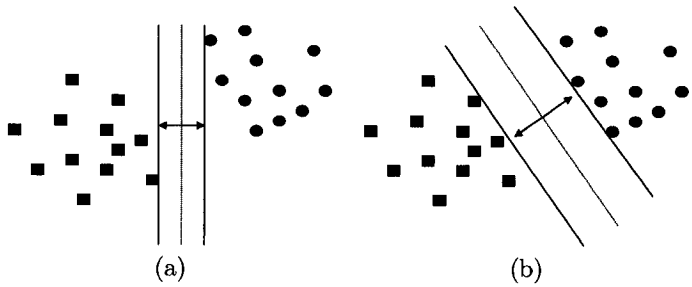


Figure 2.1: (a) A separating hyperplane with small margin (b) A separating hyperplane with large margin.

If the two classes are not separable, we can still look for a hyperplane that maximizes the margin and minimize the quantity of examples proportional to the misclassification error. The trade off between the margin and misclassification error is controlled by a constant C. In this case, the solution is a linear classifier of $D(x) = \sum_{i=1}^{N} w_i \phi(x) + b$, which satisfies

$$\begin{array}{rl} \min & \frac{1}{2} w^T w + C \sum_{i=1}^{l} \xi_i \\ \text{subject to} & y_i (\sum_{i=1}^{N} w_i \phi(x_i) + b) \geq 1 - \xi_i \\ & \xi_i \geq 0 \end{array}$$

Since it is unlikely that any real life problem can actually be solved by a linear classifier, the technique has to be extended in order to allow for non-linear decision surfaces. A method called *kernel trick* is used to convert the linear classifier into a non-linear one. This is done by mapping the original variables x into a higher-dimensional non-linear space so that linear classification in the new space is equivalent to non-linear classification in the original space. The linear function $\phi(x)$ is then replaced with a non-linear function. Actually, it is not necessary to look for the non-linear function $\phi(x)$. In the optimization process looking for the decision function of

$$\sum_{k=1}^{p} \alpha_k K(x_k, x) + b$$

is enough. In this decision function $K(x, x')$ is called the kernel function

$$K(x, x') = \sum_i \phi_i(x)\phi_i(x').$$

We can choose different kernel functions to map the original set of variables into a new space. Some commonly used kernel functions are Polynomial expressions, Gaussian functions, and multilayer perceptrons. By choosing a suitable kernel function and parameter setting, SVM can be used for real world classification tasks.

2.1.4 Artificial Neural Networks

Artificial Neural Networks (ANNs) are robust learning methods that can approximate real-valued, discrete-valued, and vector-valued target functions. For some real world applications, ANNs have been shown to be the most effective learning methods.

ANNs was partly inspired by the observation that biological learning systems are built of very complex webs of interconnected neurons. In rough analogy, ANNs are built of a set of densely interconnected simple units. Each unit takes a number of inputs and produces a single output.

One type of ANN is based on a unit called a perceptron. A perceptron takes a vector of real-valued inputs, computes a linear combination of these inputs, then outputs a 1 if the result is greater than some threshold and -1 otherwise. More specifically, given an input vector $\bar{x} = \{x_1, \cdots, x_n\}$, the output $o(\bar{x})$ computed by the perceptron is

$$o(\bar{x}) = \begin{cases} 1 & \text{if } \bar{w} \cdot \bar{x} > 0 \\ -1 & \text{otherwise} \end{cases}$$

where $\bar{w}$ is a weight vector.

This simple type of perceptron is a linear function with a limited representational power. To represent highly nonlinear functions, we use a sigmoid function to replace the simple linear output function. This unit is called a sigmoid unit, which is similar to a perceptron, but based on a smoothed and differentiable threshold function. More specifically, the sigmoid unit computes its output o as

$$o = \frac{1}{1 + e^{-\bar{w} \cdot \bar{x}}}$$

Learning a unit involves choosing values for weights w_i. One algorithm that solves the unit learning is the *delta rule* method. This method uses gradient descent to search the hypothesis space of possible weight vectors to find the weights that best fit the training examples.

When a number of units are densely interconnected to build a multilayer network, we need to use an algorithm to learn the entire network. The most commonly used learning algorithm is called backpropagation. This method employs gradient descent to attempt to minimize the squared error between the network output values and the target values for these outputs. Details of the backpropagation learning algorithm can be found in [55].

2.2 Previous Work on Measures

Although measures are quite important, there is only a few literatures investigating the characteristics of performance measures. Caruana and Niculescu-Mizil [9] empirically compared 9 commonly used machine learning measures. Flach [23] used the ROC space as the tool to theoretically compare measures. Before reviewing these works, we first give a detailed discussion for ROC space, ROC curve, and AUC (Area Under the ROC Curve).

2.2.1 ROC and AUC

The ROC (Receiver Operating Characteristics) curve was first used in signal detection theory to represent the tradeoffs between hit rates and false alarm rates [19, 24]. It has been extensively studied and applied in medical diagnosis since the 1970's [43, 61]. Spackman [59] was one of the first researchers who used the ROC graph to compare and evaluate machine learning algorithms. In recent years, extensive research on ROC has been done in machine learning [49, 50]. The area under the ROC curve, or simply AUC, provides a good "summary" for the performance of the ROC curves. Below we will provide a brief overview of ROC and its AUC.

A ROC graph is depicted in a two dimensional space. On a ROC graph, the x and y axes are plotted with the true positive rate TP and the false positive rate FP introduced in Section 1.1, respectively. In the ROC space, each classifier with a given

class distribution and cost matrix is represented by a point (TP, FP) on the ROC curve. For a model that produces a continuous output, i.e., the probability estimates for Bayesian networks, TP and FP can vary as the threshold on the output varies between its extremes (0 and 1). The resulting curve is called the ROC curve.

The ROC curve compares the classifiers' performance across the entire range of class distributions and error costs. Figure 2.2 shows a plot of four ROC curves, each representing one of the four classifiers, A through D. A ROC curve X is said to be better than 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 are both better than D.

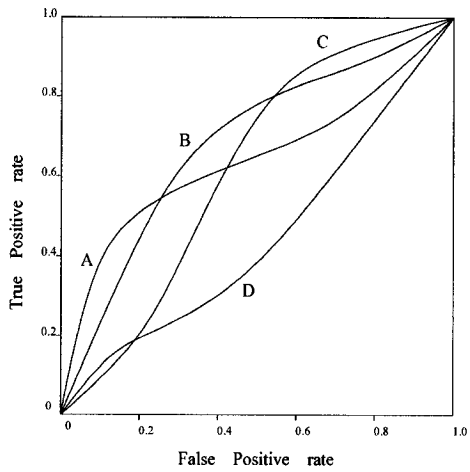


Figure 2.2: An example of four ROC curves

However, often we cannot say one curve is better than another one. For example, for curves A and B, we cannot say which one is better in the whole range. In those situations, or when the class distribution and error costs are unknown, the area under the ROC curve, or simply AUC, is a good criterion for comparing the two ROC curves. We use $AUC(X)$ to denote the area under the ROC curve X in the ROC space. In Figure 2.2 since ROC curves A and B are both better than ROC curve D, we can easily obtain that $AUC(A) > AUC(D)$ and $AUC(B) > AUC(D)$. AUC has a special statistical meaning: it represents the probability that a randomly chosen negative example will have a smaller estimated probability of belonging to the positive class than a randomly chosen positive example [26]. Moreover, AUC also equals the quantity of Wilcoxon statistic [24]. Hand and Till [25] present a simple

approach to calculating AUC of a classifier for binary classification. We will give a brief description below.

As we discussed earlier, a ROC curve is constructed by plotting different points (FP, TP) as we move the threshold t between the extreme points 0 and 1. For a specific threshold t, $TP(t)$ is the probability that a randomly chosen positive point will have a larger probability of belonging to the positive class than t. We assume that the probability density function of the probability that a randomly chosen negative point will have a larger probability of belonging to the positive class than t is $fp(t)$. The threshold t is the probability of randomly chosen negative points belonging to the positive class, we can move the threshold t to cover the whole FP distribution. Then the probability that a randomly chosen positive point will have a larger probability of belonging to the positive class than a randomly chosen negative point, which is equivalent to the probability that a randomly chosen negative point will have a smaller probability of belonging to the positive class than a randomly chosen positive point, is $\int TP(t)fp(t)dt$.

On the other hand, we can see that the area under the ROC curve is $\int TP(t)dFP(t) = \int TP(t)fp(t)dt$. Thus we can conclude that AUC is equivalent to the probability that a randomly chosen negative point will have a smaller probability of belonging to the positive class than a randomly chosen positive point [25].

calculating AUC of a classifier are n_0 positive examples and n_1 negative examples. We rank all these $n_0 + n_1$ examples incrementally according to their probabilities of belonging to the positive class. Assume that the ith positive example is ranked as the r_ith example in the ranked list. Then there are $r_i - i$ negative examples that have smaller probabilities of belonging to the positive class than that of the ith positive example. Since there are totally n_1 negative examples, the probability of a randomly chosen negative example that has lower probability of belonging to positive class than that of the ith positive example is $\frac{r_i - i}{n_1}$. Since the probability of choosing the ith positive example when we randomly choose a positive example is $\frac{1}{n_0}$, the probability that a randomly chosen negative example will have a smaller estimated probability of belonging to the positive class than a randomly chosen positive example is

$$\sum_{i=1}^{n_0} \frac{1}{n_0} \frac{(r_i - i)}{n_1} = \frac{\sum r_i - \sum i}{n_0 n_1} = \frac{S_0 - n_0(n_0+1)/2}{n_0 n_1},$$

This leads to the AUC formula 1.1 that we have introduced in of Chapter 1.

$$AUC = \frac{S_0 - n_0(n_0 + 1)/2}{n_0 n_1}$$

2.2.2 Comparing Measures Empirically

Caruana and Niculescu-Mizil [9] conducted a significant research by empirically comparing some performance measures. They used a variety of learning methods to compare nine boolean classification performance measures: Accuracy, lift, F-measure, AUC, APR, BEP, RMS, MXE, and Probability Calibration. They first performed a Multidimensional scaling (MDS) analysis and used the space graphs to show the relationships among measures. They showed that the three probability measures, RMS, MXE, and calibration, lay in one part of measure space far away from the ranking measures of AUC, APR, BEP, and lift. In between them fall two threshold measures of accuracy and F-measure. They also empirically computed the rank correlations among different measures by using learning models that run on some large real-world datasets. They demonstrated that some measures are closely correlated (with rank correlations above 0.90), such as AUC with lift, AUC with APR, RMS with MXE, accuracy with BEP, and so on. To deal with the problem that under a certain situation it is not known how to choose the best measure, they also proposed to design a new measure: SAR. SAR combines RMS, accuracy, and AUC into one measure. SAR is shown to be very robust and is expected to perform well in general situations. Finally, they also evaluated the performance of learning algorithms on different measures. They showed that SVMs and boosted trees have excellent performance on measures like accuracy, but perform poorly on probability measures such as RMS. They also showed that it is unexpected that SVMs and boosted trees have excellent performance on ranking measures such as AUC and APR.

In this thesis we also empirically compare some performance measures. However, our comparisons are different. We do not attempt to compute the correlations among different measures. Instead we will establish a framework to compare measures in Chapter 3. Based on this framework, some popular measures are empirically compared.

2.2.3 Comparing Measures in ROC Space

Flach [23] theoretically compared some measures by using ROC space as the tool. As discussed in Section 2.2.1, a classifier can be depicted as a curve in ROC space. However, a measure cannot be simply depicted as a curve. For a measure with a fixed value, it can correspond to many points in the ROC space. The collections of all points that correspond to a fixed measure value constitute a series of curves, which are called isometrics. Flach [23] explored the characteristics of some commonly used measures of accuracy, precision, recall, F-measure, and decision tree splitting criteria of Entropy, Gini by depicting their isometrics. He compared the measures with their effective slopes of isometrics. He claimed that a measure's characteristic is determined by the slope of its isometrics at any point in ROC space. He obtained some interesting results, such as designing a simplification version of F-measure, deriving a new decision tree splitting criterion Gini-split that is insensitive to class skews.

In this thesis, we will introduce a different framework to study the relationships among some commonly used measures. We will focus on studying the consistency relationship between two measures, and whether one measure has more discriminatory power than the other.

Chapter 3

Comparing Machine Learning Measures

In this chapter we establish a general framework to compare measures of machine learning. We propose a set of criteria to provide detailed and complete comparisons between measures. Based on these criteria, we first compare two popular measures: accuracy and AUC, and we then compare some ranking measures. These comparisons are very useful in evaluating and constructing learning algorithms. Based on the comparison results, we show that generally AUC is better than accuracy in evaluating learning algorithms. We also give a preference order in selecting ranking measures to evaluate ranking performance.

3.1 Criteria for Comparing Measures

We first propose five percentage criteria and two degree criteria to explore and compare the detailed difference in predictive performance for two arbitrary measures. The new criteria can answer detailed questions such as how much the two measures are consistent, inconsistent, and how much one measure is more discriminant than the other one.

We first discuss the equivalence of two measures. We assume that f and g are two functions mapping into a total order that represents an evaluation measures on elements in a domain Ψ, and a and b are two elements in Ψ. Intuitively we say f and g

are *equivalent* whenever f stipulates that a is better than b, if and only if g stipulates that a is better than b. For example, when using two scoring systems (measures) to evaluate students, if one scoring system uses A, B, C, D, and F, which correspond to Excellent, Good, Average, Pass, and Fail respectively in another scoring system, then clearly these two scoring systems are equivalent. When two measures are equivalent, there is a one-to-one mapping between the two measures with the same order.

Another useful notion in comparing two measures is the dominance relation. Intuitively, we say that f dominates g if and only if whenever f stipulates that a is better than b, g stipulates that a is better than or equal to b. As an example, let us consider the numerical marks and letter marks (the two measures) that evaluate university students (the domain). Normally, the letter mark A corresponds to numerical marks from 80 to 100, B from 70 to 79, C from 60 to 69, D from 50 to 59, and F from 0 to 49. If we use f for the numerical mark and g for the letter mark, then for any two students a and b, if $f(a) > f(b)$ then $g(a) \geq g(b)$. Therefore the numerical marks dominate letter marks.

These intuitions can be made precise in the following definitions.

Definition 1 (Equivalence) Two measures f and g are *equivalent*, if for any a, $b \in \Psi$ $f(a) > f(b)$ iff $g(a) > g(b)$.

Definition 2 (Dominance) A measure f dominates another measure g if for any a, $b \in \Psi$, $f(a) > f(b)$ implies $g(a) \geq g(b)$. In addition, there exist a, $b \in \Psi$, $f(a) > f(b)$ and $g(a) = g(b)$.

Unfortunately, most evaluation measures in machine learning are not equivalent. For two arbitrary measures, one measure usually does not dominate the other one. For example, although it can be shown easily that AUC takes more values than accuracy, AUC and accuracy are not analogous to numerical marks and letter marks. Sometimes AUC and accuracy are inconsistent with, or contradictory to each other, and therefore, AUC does not dominate accuracy. This can be seen in the following example: Table 3.1 lists two ranked lists of 10 testing examples, presumably as the result of predictions from two learning algorithms. The AUC of the ranked list a is $\frac{21}{25}$, and the AUC of the ranked list b is $\frac{16}{25}$. Thus the ranked list a is better than the ranked list b according to AUC. Assuming that both learning algorithms classify half (the right most 5) of the 10 examples as positive, and the other 5 as negative,

the accuracy of a is 60%, and the accuracy of b is 80%. Therefore, b is better than a according to accuracy. This example shows that there exist cases where AUC and accuracy are inconsistent, and thus, AUC and accuracy do not dominate each other.

Table 3.1: A counter example in which AUC and accuracy are inconsistent.

a	$-$	$-$	$-$	$+$	$+$	$\|$	$-$	$-$	$+$	$+$	$+$
b	$+$	$-$	$-$	$-$	$-$	$\|$	$+$	$+$	$+$	$+$	$-$

For two measures that are not consistent in the whole domain (such as AUC and accuracy in the example above), we can define percentages of consistency and inconsistency, which give precisely the probabilities of two measures being consistent and inconsistent in comparing two different elements in the domain. Assume that T is the total number of pairs formed by two different elements in Ψ, we have:

Definition 3 (Percentage of Consistency) Let $R = \{(a,b)|a,b \in \Psi, f(a) > f(b), g(a) > g(b)\}$, the percentage of consistency of f and g is defined as $CON_{f,g} = \frac{|R|}{T}$.

Definition 4 (Percentage of Inconsistency) Let $R = \{(a,b)|a,b \in \Psi, f(a) > f(b), g(a) < g(b)\}$, The percentage of inconsistency of f and g is defined as $INCON_{f,g} = \frac{|R|}{T}$.

Note that the sum of two percentages defined above is usually less than one, as there are additional situations when comparing two measures, as discussed below. In the example of numerical and letter marks, there are many cases in which numerical marks (such as 91 and 98) can tell the difference when letter marks cannot (both 91 and 98 correspond to A). The reverse cannot be true; there is no case where letter marks can tell the difference but numerical marks cannot. That is, numerical marks are strictly more discriminant than letter marks. However, this analogy cannot be carried over to AUC and accuracy. There are many cases in which AUC can tell the difference between two ranked lists but accuracy cannot, but counter examples also exist in which accuracy can tell the difference but AUC cannot. Table 3.2 shows such a counter example. We can see that both ranked lists have the same AUC ($\frac{3}{5}$) but different accuracies (60% and 40% respectively).

Thus we can define percentage of discriminancy for f over g and for g over f as follows.

Table 3.2: A counter example in which two ranked lists have same AUC but different accuracies

a	−	−	+	+	−	+	+	−	−	+
b	−	−	+	+	+	−	−	+	−	+

Definition 5 (Percentage of Discriminancy) Let $P = \{(a,b)|a,b \in \Psi, f(a) > f(b), g(a) = g(b)\}$, and $Q = \{(a,b)|a,b \in \Psi, g(a) > g(b), f(a) = f(b)\}$. The percentage of discriminancy for f over g is defined as $DIS_{f/g} = \frac{|P|}{T}$, and the percentage of discriminancy for g over f is defined as $DIS_{g/f} = \frac{|Q|}{T}$.

The last situation is for cases where neither of the two measures can tell the difference. Table 3.3 illustrates two ranked lists with the same AUC (3/5) and accuracy (60%). Thus, we can define Percentage of Indifferency below.

Table 3.3: An example in which neither AUC nor accuracy can tell the difference between two ranked lists.

a	−	−	+	+	−	+	+	−	−	+
b	−	−	+	+	−	+	−	+	+	−

Definition 6 (Percentage of Indifferency) Let $R = \{(a,b)|a,b \in \Psi, f(a) = f(b), g(a) = g(b)\}$, the percentage of indifferency of f and g is defined as $IND_{f,g} = \frac{|R|}{T}$.

Figure 3.1 gives an illustration of the five "percentage criteria" for comparing two measures f and g. Each slice is one "percentage criterion". The slice with the symbol ">,>" means that "$f(a) > f(b), g(a) > g(b)$", thus it represents the percentage of consistency. Other slices and symbols represent the other criteria correspondingly. From this figure we can see that the whole space of the comparison can be partitioned into the five regions according to our five "percentage criteria", and the sum of the five percentages is equal to 1.

Clearly, for a measure f to be "better" than g, f and g must be more likely to be consistent than inconsistent. That is, $CON_{f,g} > INCON_{f,g}$. In addition, f should

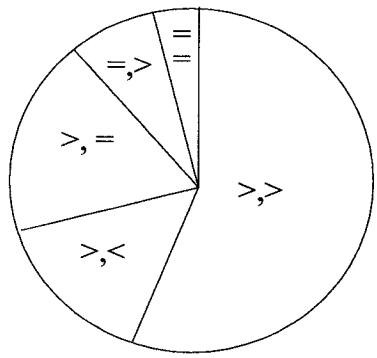


Figure 3.1: Illustrations of the five percentage criteria.

more often be more discriminant over g than g over f. That is, $DIS_{f/g} > DIS_{g/f}$. To view this visually in figure 3.1, the area of "$>,>$" should be larger than the area of "$>,<$", and the area of "$>,=$" should be larger than the area of "$=,>$".

Naturally we can define two more criteria that can be directly used to study whether two measures are statistically consistent, and whether one measure is more discriminant than the other. These two criteria are degree of consistency and degree of discriminancy respectively. They can be derived in terms of CON, INCON, and DIS.

Definition 7 (Degree of Consistency) For two measures f and g, the degree of consistency is defined as $C_{f,g} = \frac{CON_{f,g}}{CON_{f,g}+INCON_{f,g}}$.

Definition 8 (Degree of Discriminancy) For two measures f and g, the degree of discriminancy for f over g is defined as $D_{f,g} = \frac{DIS_{f/g}}{DIS_{g/f}}$.

The concept of percentage of indifferency (IND) between f and g in Definition 6 can also be viewed as the degree of frequency that f and g both cannot discriminate. For completeness, we also define the degree of indifferency as follows

Definition 9 (Degree of Indifferency) For two measures f and g, the degree of indifferency of f and g is defined as $E_{f,g} = IND_{f,g}$.

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 using the above two 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. Note that the notion that f is a better measure than g is not based on some subjective evaluation that f is closer to some true target measure than g; instead, it is based on the objective criteria of consistency and discriminancy between the two measures f and g themselves.

Definition 10 A measure f is statistically consistent and more discriminating than g, or intuitively, f is a better measure than g, if and only if $\mathbf{C_{f,g}} > 0.5$, and $\mathbf{D_{f/g}} > 1$. We denote this by $f \succ g$.

The five percentage criteria and three degree criteria proposed here provide a refined and detailed comparison between two arbitrary single-number measures. They allow us to study precisely how much (the probability) of two measures that are consistent, inconsistent, indifferent, and how much one measure is more discriminating than the other. This framework can be applied to compare any single-number measures in machine learning, other experimental science, and engineering areas. They also allow us to construct new measures that are better than the existing ones.

In the next section we will apply these criteria to give a thorough comparison of two most commonly used machine learning measures: accuracy and AUC.

3.2 Comparing Accuracy and AUC

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 decision trees [51] and Naive Bayes [18]) 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, we often need to promote the top X% (X can be 5 or 10) 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 [40], 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 [12]. 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 answer lies in the ROC curve.

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]. Recently, other researchers have even used AUC to construct learning algorithms [22, 39]. But it is not clear if and why AUC is a better measure than accuracy.

In the next subsections, we will apply the comparison criteria proposed in the previous section and show, both formally and empirically, that AUC is a better measure than accuracy. Our result suggests that AUC should replace accuracy in comparing learning algorithms in the future. Our result also prompts 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

when compared by accuracy [35, 36, 16]. Since we will establish that AUC is a better measure, are those learning algorithms still very similar when compared by AUC? How does recent Support Vector Machine (SVM) [6, 14, 56] 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 3.2.4.

Conclusions drawn in this section may spur new research in machine learning. As a new measure (such as AUC) is discovered and proved to be better than a previous measure (such as accuracy), we can re-design most learning algorithms to optimize the new measure [22, 39]. 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 [39]. Results in this section suggest that in real-world applications of machine learning and data mining we should use learning algorithms to optimize AUC instead of accuracy. Most learning algorithms today still optimize accuracy directly (or indirectly through entropy, for example) as their goals.

3.2.1 Theoretical Comparison

We first give a theoretical comparison between AUC and accuracy. We will formally prove that AUC is consistent to, and more discriminant measure than, accuracy. We substitute AUC for f, and accuracy for g in the definitions of degree of consistency and discriminancy. To simplify our notation, we will use AUC to represent AUC values, and acc for accuracy. The domain Ψ consists, in general, of ranked lists of testing examples. In the Theorems below, however, we restrict the domain Ψ to be binary (with two classes) ranked lists in Theorem 1, and we restrict the domain Ψ to be binary, balanced (with the same number of positive and negative examples) ranked lists in Theorem 2. Since we require $\mathbf{C} > 0.5$ and $\mathbf{D} > 1$ we will need to prove:

Theorem 1 Given a domain Ψ of all possible 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 $|P| > |Q|$.

Without loss of generality, we assume that there are n_0 positive examples and n_1 negative examples in any ranked list, and we set the cutoff point between the n_1th example and the (n_1+1)th example (the classifier classifies n_0 examples as positive and other n_1 examples as negative). For brevity in the following discussion we use the notation AUC_{max}, AUC_{min} to denote the maximum and minimum AUC values in a specific context. For example, $AUC_{max}(\alpha)$ is the maximum AUC value that any ranked list with accuracy α can reach, and $AUC_{max}(r)$ is the maximum AUC value that any ranked list r can reach. Lemma 5 is the basis for proving theorem 1. To prove lemma 5, we propose and prove lemma 1 to lemma 4.

Lemma 1 *For a given ranked list r with an accuracy α,*
$AUC_{min}(r) = \frac{(n_0+n_1)^2\alpha^2-(n_0-n_1)^2}{4n_0n_1}$, $AUC_{max}(r) = 1 - \frac{(n_0+n_1)^2}{4n_0n_1}(1-\alpha)^2$.

Proof. Assume that there are n_p positive examples correctly classified. Then there are $n_0 - n_p$ positive examples in the negative section. When all $n_0 - n_p$ positive examples are put on the highest positions in the negative section and n_p positive examples are put on the highest positions in the positive section, AUC reaches the maximum value. For each positive example in the negative section, there are $(n_1 - n_0 + n_p)$ negative examples ranked lower than it. And for each positive example in the positive section, there are n_1 negative examples ranked lower than it. Therefore

$$AUC_{max}(\alpha) = \frac{\sum_{i=1}^{n0-n_p}(n_1 - n_0 + n_p) + \sum_{i=n_0-n_p+1}^{n_0} n_1}{n_0n_1} = \frac{n_1n_p + (n_0 - n_p)(n_1 - n_0 + n_p)}{n_0n_1}.$$

Similarly when all positive examples are put on the lowest positions in both the positive and negative sections, AUC reaches the minimum value.

$$AUC_{min}(\alpha) = \frac{n_p(n_1 - n_0 + n_p)}{n_0n_1}$$

Since $\alpha = \frac{2n_p+n_1-n_0}{n_0+n_1}$, from above two formulas, we can obtain $AUC_{min} = \frac{(n_0+n_1)^2\alpha^2-(n_0-n_1)^2}{4n_0n_1}$, $AUC_{max} = 1 - \frac{(n_0+n_1)^2}{4n_0n_1}(1-\alpha)^2$. □

Lemmas 2 and 3 can be directly derived from Lemma 1.

Lemma 2 *For two given ranked lists r and s with accuracies α and β respectively, if $\alpha > \beta$, then $AUC_{max}(r) > AUC_{max}(s)$, $AUC_{min}(r) > AUC_{min}(s)$.*

Proof. From Lemma 1, we have $AUC_{max}(r) = 1 - \frac{(n_0+n_1)^2}{4n_0n_1}(1-\alpha)^2$, $AUC_{max}(s) = 1 - \frac{(n_0+n_1)^2}{4n_0n_1}(1-\beta)^2$, since $\alpha > \beta$, we have $AUC_{max}(r) > AUC_{max}(s)$. Similarly we have $AUC_{min}(r) > AUC_{min}(s)$. □

Lemma 3 *For any ranked list r with accuracy α, the number of different AUC values that r can reach is* $\frac{(\alpha-\alpha^2)(n_0+n_1)^2}{2} + 1$.

Proof. Since the difference of two adjacent AUC values is $\frac{1}{n_0n_1}$, the total number of different AUC values is $\frac{AUC_{max}(\alpha)-AUC_{min}(\alpha)}{\frac{1}{n_0n_1}} + 1 = \frac{(\alpha-\alpha^2)(n_0+n_1)^2}{2} + 1$. □

Lemma 4 *Let R and S be two sets of ranked lists. $R = \{r|acc(r) = \alpha, AUC(r) = \beta\}$, $S = \{r|acc(r) = \alpha, AUC(r) = AUC_{max}(\alpha) + AUC_{min}(\alpha) - \beta\}$. Then $|R| = |S|$.*

Proof. For any ranked list $r \in R$, we perform the following position switch to obtain another ranked list r'. For any positive example in the negative section whose position is r_i, we switch it with the example in the position of $n_1 + 1 - r_i$. For any positive example in the positive section whose position is r_i, we switch it with the example in the position of $n_0 - (r_i - n_1) + 1 + n_1$. These position switches put all the positive examples in the positive section into negative section, and put all the positive examples in the negative section into positive section. Suppose that there are n_p positive examples in r's positive section. Thus $acc(r') = \frac{n_0-n_p+n_1-n_p}{n_0+n_1} = acc(r)$. From formula 1.1 we have

$$AUC(r') = \frac{\sum_{i=n_0-n_p+1}^{n_0}(n_1+1-r_i) + \sum_{i=1}^{n_0-n_p}(n_0-(r_i-n_1)+1+n_1)}{n_0n_1}.$$

This can be simplified to

$$AUC(r') = AUC_{max}(\alpha) + AUC_{min}(\alpha) - \beta$$

Thus we have $r' \in S$. Clearly, the position switches make different $r \in R$ correspond to different r', and vice versa. Therefore the mapping $r \mapsto r'$ is a a one-to-one mapping from R to S. $|R| = |S|$. □

Lemma 5 is the direct basis of theorem 1.

Lemma 5 *Suppose accuracies $\alpha > \beta$. Let $P = \{(a,b)|acc(a) = \alpha, acc(b) = \beta, AUC(a) > AUC(b)\}$, $Q = \{(a,b)|acc(a) = \alpha, acc(b) = \beta, AUC(a) < AUC(b)\}$. Then $|P| > |Q|$.*

Proof. 1. If $AUC_{min}(\alpha) \geq AUC_{max}(\beta)$, then it is obvious that $|P| > 0$, $|Q|$=0, $|P| > |Q|$. 2.Otherwise, there are two cases. 1) If $AUC_{max}(\beta) \leq (AUC_{max}(\alpha) + AUC_{min}(\alpha))/2$. Let $AUC_{max}(\beta) \geq \gamma_1 > \gamma_2 \geq AUC_{min}(\alpha)$. For any ranked lists r, s with $acc(r) = \alpha$, $AUC(r) = \gamma_2$, $acc(s) = \beta$, $AUC(s) = \gamma_1$ we have $(r, s) \in Q$. For any ranked list r' with $acc(r') = \alpha$, $AUC(r') = AUC_{max}(\alpha) + AUC_{min}(\alpha) - \gamma_2$ we have $(r', s) \in P$. By lemma 4 the number of r' equals to the number of r. Thus the number of pairs (r', s) equals to the number of pairs (r, s). Since it is easy to obtain that the mapping $(r, s) \mapsto (s', r)$ is injective, we can obtain $|P| > |Q|$. 2) For $AUC_{max}(\beta) > (AUC_{max}(\alpha) + AUC_{min}(\alpha))/2$. Similar to 1) we can also prove $|P| > |Q|$. □

Theorem 1 shows that the consistency of accuracy and AUC for binary datasets (balanced or imbalanced) is greater than 0.5.

Theorem 1 *Given a domain Ψ of all possible 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 $|R| > |S|$.*

Proof. Let $R_{\alpha\beta} = \{(a, b)|AUC(a) > AUC(b), acc(a) = \alpha, acc(b) = \beta\}$, $S_{\alpha\beta} = \{(a, b)|AUC(a) < AUC(b), acc(a) = \alpha, acc(b) = \beta\}$, and suppose $\alpha > \beta$.

Clearly, $R = \bigcup_{\alpha,\beta} R_{\alpha\beta}$, $S = \bigcup_{\alpha,\beta} S_{\alpha\beta}$, and $R_{\alpha_1\beta_1} \cap R_{\alpha_2\beta_2} = \phi$, $S_{\alpha_1\beta_1} \cap S_{\alpha_2\beta_2} = \phi$, for $\alpha_1 \neq \alpha_2$ or $\beta_1 \neq \beta_2$. So $|R| = \sum_{\alpha,\beta} |R_{\alpha\beta}|$, $|S| = \sum_{\alpha,\beta} |S_{\alpha\beta}|$. By lemma 5, we have $|R_{\alpha\beta}| > |S_{\alpha\beta}|$. Therefore $|R| > |S|$. □

For the discriminancy between AUC and accuracy, we only study the case where the ranked lists contain an equal number of positive and negative examples, and we assume that the cutoff for classification is at the exact middle of the ranked list (each classifier classifies exactly half examples into positive class and the other half into negative class). So in the following lemmas and theorems, we assume $n_0 = n_1$ and we always use n instead of n_0 and n_1. If we use k_i to represent the number of negative examples ranked lower than the ith positive example, then it is easy to obtain that $k_i = r_i - i$. We use the notation $\sigma(r) = \sum k_i$, and $AUC = \frac{\sigma(r)}{n^2}$. Furthermore we use $|r|_+$ to represent the number of positive examples in ranked list r. Lemma 6 to lemma 9 are proposed as the basis to prove theorem 2.

Lemma 6 *Let R_1 and R_2 be two ranked lists sets, and $R_1 = \{r||r| = n, |r|_+ = m, 0 < m < \frac{n}{2}, \sigma(r) = C\}$, $R_2 = \{r||r| = n, |r|_+ = m + 1, 0 < m < \frac{n}{2}, \sigma(r) = C\}$. Then $|R_2| \geq |R_1|$.*

Proof. This lemma is related with the restricted partitions in *Number Theory* [1]. Let $p(N, M, n)$ denote the number of partitions of positive integer n into at most M parts, each $\leq N$. It is easy to obtain that $|R_1| = p(n-m, m, C)$, $|R_2| = p(n-m-1, m+1, C)$. So we need to prove $p(n-m-1, m+1, C) \geq p(n-m, m, C)$, for $m < n/2$. This is the result of [2]. □

Lemma 7 *Let R_1 and R_2 be two ranked lists sets. $R_1 = \{r||r| = n, |r|_+ = m, 0 < m < \frac{n}{2}, \sigma(r) = C\}$, $R_2 = \{r||r| = n, |r|_+ = n-m, 0 < m < \frac{n}{2}, \sigma(r) = C\}$. Then $|R_1| = |R_2|$.*

Proof. For any ranked list $r \in R_1$, we switch the example on position i with the example on position $n-i$ to obtain another ranked list s. Since $AUC(r) = \frac{\sum_{i=1}^{m} r_i - m(m+1)/2}{m(n-m)}$, $AUC(s) = \frac{\sum_{i=1}^{m}(n-r_i) - m(m+1)/2}{m(n-m)}$. We then replace all positive and negative examples in s with negative and positive examples respectively to obtain ranked list s'. It is easy to obtain $AUC(s') = AUC(r)$, $|s'|_+ = n - m$. Thus $s' \in R_2$. Therefore $|R_1| \leq |R_2|$. Similarly we can prove $|R_2| \leq |R_1|$. We have $|R_1| = |R_2|$. □

Lemma 8 *Let $\alpha \geq \frac{1}{2} \geq \beta$, and $\alpha + \beta = 1$. R_1 and R_2 are two ranked lists sets. $R_1 = \{r|acc(r) = \alpha, AUC(r) = AUC_{min}(\alpha) + \gamma\}$, $R_2 = \{s|acc(s) = \beta, AUC(s) = AUC_{min}(\beta) + \gamma\}$, Then $|R_1| = |R_2|$.*

Proof. For any ranked list $r \in R_1$, from the definition of σ, $\sigma(r) = AUC_{min}(\alpha)n^2 + \gamma n^2$. Let the positive section of r be ranked list r_1 and the negative section of r be ranked list r_2, then $r = r_2 r_1$. Since $\sigma(r) = \sigma(r_2) + \sigma(r_1) + n_p(n_1 - n_0 + n_p) = \sigma(r_2) + \sigma(r_1) + AUC_{min}(\alpha)\gamma n^2$, We have $\sigma(r_1) + \sigma(r_2) = \gamma n^2$. We then construct a new ranked list s by combining r_2 and r_1 such that $s = r_1 r_2$. Then $acc(s) = 1 - \alpha = \beta$, $AUC(s) = AUC_{min}(\beta) + \gamma$. So $s \in R_2$, $|R_1| \leq |R_2|$. Similarly we can prove $|R_2| \leq |R_1|$. Therefore $|R_1| = |R_2|$. □

Lemma 9 *Let accuracies α and β satisfy one of the following conditions.*
1)$\frac{1}{2} \geq \alpha > \beta$, or
2)$\beta > \alpha \geq \frac{1}{2}$, or
3)$\alpha > \frac{1}{2} > \beta$ and $1 - \alpha \geq \beta$, or
4)$\beta > \frac{1}{2} > \alpha$ and $1 - \alpha < \beta$.
Let $R_1 = \{r|acc(r) = \alpha, AUC(r) = AUC_{min}(\alpha) + \gamma\}$, $R_2 = \{s|acc(s) = \beta, AUC(s) = AUC_{min}(\beta) + \gamma\}$. Then $|R_1| \geq |R_2|$.

Proof. We first consider the case that α and β satisfy condition 1). For any ranked list $s \in R_2$ we split the positive and negative sections of s into ranked lists s_1 and s_2. Then $|s_1| = |s_2|$, $\sigma(s_1) + \sigma(s_2) = \gamma n^2$. We extract a subset $T_i \subseteq R_2$, $T_i = \{s | s = s_2 s_1, |s_1| = |s_2|, acc(s) = \beta, \sigma(s_1) = i, \sigma(s_2) = \gamma n^2 - i\}$. We also extract a subset $P_i \subseteq R_1$, $P_i = \{r | r = r_2 r_1, |r_1| = |r_2|, acc(r) = \alpha, \sigma(r_1) = i, \sigma(r_2) = \gamma n^2 - i\}$.

Since $\frac{1}{2} \geq \alpha > \beta$, for any $r = r_2 r_1 \in P_i$, $s = s_2 s_1 \in T_i$, $|s_1|_+ \leq |r_1|_+ \leq \frac{|r_1|}{2}$. By lemma 6, there are more ranked lists r_1 than s_1. By lemma 7, the number of ranked lists r_2 equals to the number of ranked lists r_2' which satisfy $|r_2'|_+ = n - |r_2|_+$, and $\sigma(r_2') = \gamma n^2 - i$; the number of ranked lists s_2 equals to the number of ranked lists s_2' which satisfy $|s_2'|_+ = n - |s_2|_+$, and $\sigma(s_2') = \gamma n^2 - i$. Since $|r_2'|_+ \geq |s_2'|_+$, there are more ranked lists r_2 than s_2. Therefore $|P_i| \geq |T_i|$. Sum these inequalities for all i we can obtain $|R_1| \geq |R_2|$. Similarly we can prove cases 2), 3), 4) from case 1) and lemma 8. □

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 $|P| > |Q|$.*

Proof. Let $A_{\alpha,\gamma} = \{r | acc(r) = \alpha, AUC(r) = \gamma\}$, $B_{\beta,\gamma} = \{s | acc(s) = \beta, AUC(s) = \gamma\}$. Suppose α and β satisfy one of the conditions in lemma 9. Let $C_{\alpha,\beta,\gamma} = \{r | acc(r) = \alpha, AUC(r) = \gamma - AUC_{min}(\beta) + AUC_{min}(\alpha)\}$. By lemma 9, we have $|C_{\alpha,\beta,\gamma}| \geq |B_{\beta,\gamma}|$. Thus $\sum_{\alpha,\beta,\gamma} |A_{\alpha,\gamma}| \cdot |C_{\alpha,\beta,\gamma}| \geq \sum_{\alpha,\beta,\gamma} |A_{\alpha,\gamma}| \cdot |B_{\beta,\gamma}|$. On the other hand $|Q| = \sum_{\alpha,\beta,\gamma} |A_{\alpha,\gamma}| \cdot |B_{\beta,\gamma}|$, and it is obvious that $|P| > \sum_{\alpha,\beta,\gamma} |A_{\alpha,\gamma}| \cdot |C_{\alpha,\beta,\gamma}|$. Therefore $|P| > |Q|$. □

Theorems in Section 3.2.1 state that AUC is statistically consistent to, and more discriminating than, accuracy with binary, balanced datasets. In the following subsections, we perform extensive experiments on AUC and accuracy on a variety of artificial datasets and real-world datasets. This is necessary for three 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. This is necessary because most real-world datasets are imbalanced with multiple class values. Second, empirical experiments on artificial and real-world datasets will give us intuitions on the ranges of the degree of consistency **C**, the degree of discriminancy **D**, and the degree of indifferency **E**, on different types of datasets. Third, our theorems and empirical evaluations on artificial datasets are

based on the uniform distribution of examples. In real-world datasets, examples are often non-uniform. The experiments in Section 3.2.4 will directly evaluate the relations between AUC and accuracy on real-world datasets.

3.2.2 Comparison with Artificial Datasets

We empirically compare AUC and accuracy with artificial datasets. We will use three kinds of artificial datasets – binary balanced, binary imbalanced, and multiclass – in our experiments.

3.2.2.1 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, we still perform empirical experiments in order to gain an intuition on the ranges of the degree of consistency **C**, the degree of discriminancy **D**, and the degree of indifferency **E**.

To calculate **C**, **D**, and **E**, we exhaustively search all possible ranked lists of the same length. Since the number of ranked lists increases exponentially with the size of the ranked lists, we will only use small datasets in our experiments. 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 the degree of consistency (see Definition 7), 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 the degree of discriminancy (see Definition 8), 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 3.4 and 3.5 show the experiment results. For consistency, we can see (Table 3.4) 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 3.5 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 3.4: Experiments on statistical consistency between AUC and accuracy for the balanced binary 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 3.5: Experiments showing AUC is statistically more discriminating than accuracy for the balanced binary 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 confirm 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 3.6. 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 $1 - E$), AUC and accuracy are not indifferent; that is, they are either consistent, inconsistent, or one is more discriminant than another.

Table 3.6: 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

3.2.2.2 Imbalanced Datasets

We extend our previous results on the balanced datasets with binary classes to imbalanced datasets. We will experimentally confirm 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). For accuracy, we must decide the cut-off point. We assume that the class distributions of training and testing examples are the same because this is the fundamental hypothesis in machine learning for performance evaluation. Thus, the cut-off point of the ranked list is at the 75% position: the lower 75% of the ranked testing examples are classified as negative, and the top 25% of the ranked testing examples are classified as positive. Tables 3.7 and 3.8 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. Compared to the results for the balanced datasets (Tables 3.4 and 3.5), we can see that the degree of consistency is lower but the degree of discriminancy is higher when datasets are imbalanced.

Table 3.7: Experiments on statistical consistency between AUC and accuracy for the imbalanced binary datasets

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

Table 3.8: Experiments showing AUC is statistically more discriminating than accuracy for the imbalanced binary datasets

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

We have also obtained the degree of indifferency for the imbalanced binary datasets as shown in Table 3.9. Compared to the results in Table 3.6, we can conclude that the degree of indifferency is basically the same.

To see the effect of the class distribution to the degree of imbalanced consistency and discriminancy, we fix the number of the testing examples as 10, and vary the number of positive examples as 5 (balanced), 6, 7, 8, and 9. Table 3.10 shows the changes of consistency and discriminancy with different class distribution. As we can see, except for the extreme cases at the two ends, that 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

Table 3.9: Experimental results for the degree of indifferency between AUC and accuracy for the imbalanced binary datasets

#	$AUC(a) = AUC(b)$ & $acc(a) = acc(b)$	(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

Table 3.10: Experimental results for showing the variation of degree of consistency and discriminancy with different class distribution for binary datasets

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	∞

intuitions on degree of consistency and discriminancy in the binary datasets with different class distributions.

3.2.2.3 Multiclass Datasets

In previous experiments we only study AUC and accuracy for datasets with two (binary) classes. It is much more complicated for multiple classes cases and there is no straightforward approach to extend the definition of AUC from the case of two classes. Hand and Till [25] proposed a simple generalization of AUC for multiple classes, as follows. For a ranked list of c classes, each example has a label indicating the class it actually belongs to. Each example is assigned to 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 value p_i, and we calculate the AUC value as $AUC(i, j)$. Then we

Table 3.11: An example for calculating AUC for multiple classes.

class	1	1	2	2	3	3
p_1	0.6	0.15	0.3	0.45	0.1	0.8
p_2	0.15	0.3	0.5	0.25	0.2	0.05
p_3	0.25	0.55	0.2	0.3	0.7	0.15

sort them incrementally by the probability value p_j, 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 $\frac{2}{c(c-1)}\sum_{i<j}\hat{AUC}(i,j)$ [25]. Table 3.11 gives an example for calculating AUC of multiple classes. The ranked list has 6 examples belonging to 3 classes, and there are 2 examples for each class. The first row in Table 3.11 represents the class labels for each example. The second, third and fourth rows are the predicted probabilities for each example belonging to the 3 classes. From this table we can obtain $AUC(1,2) = \frac{1}{2}$, $AUC(2,1) = \frac{3}{4}$, $AUC(1,3) = \frac{1}{2}$, $AUC(3,1) = \frac{1}{2}$, $AUC(2,3) = 1$, $AUC(3,2) = \frac{1}{2}$. The AUC for this ranked list is then $\frac{(\frac{1}{2}+\frac{3}{4})/2+(\frac{1}{2}+\frac{1}{2})/2+(1+\frac{1}{2})/2}{3} = \frac{5}{8}$.

To perform experiments with artificial datasets for multiple classes (balanced only), 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 which covers all possible ranked lists 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.

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 to one of the c classes. This assumption is not restrictive as any ranked list is a permutation of this one.

Table 3.12 shows the experimental results for the consistency and discriminancy of the multiclass datasets. The number of classes ranges from 3 to 10, and there are 2 examples for each class. From these experimental results, we can plot the the degree

of consistency and the degree of discriminancy as functions of the number of classes, which are show in Figure 3.2. We can clearly see that when the number of classes increases, the degree of consistency decreases (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 3.12: Experiments on the consistency and discriminancy between AUC and accuracy for multiclass datasets

# of classes	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

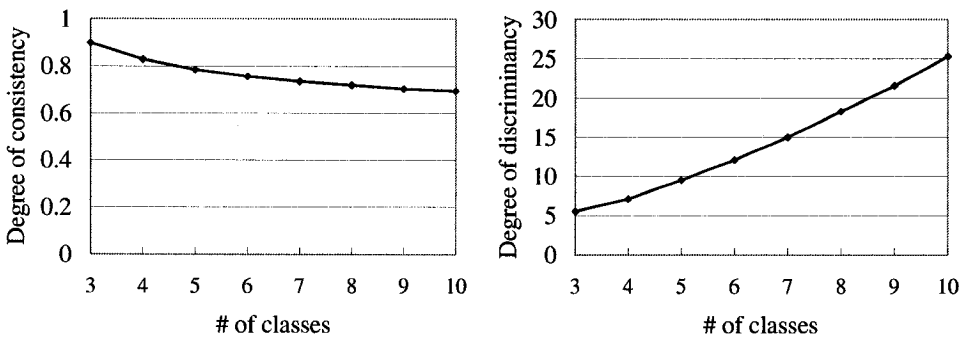


Figure 3.2: The degree of consistency (**C**) and degree of discriminancy (**D**) depicted as functions of the number of classes, from experimental results with multiclass datasets.

To conclude, for both balanced or imbalanced, binary or multiclass (3 to 10 classes) datasets, our experiments suggest that AUC is statistically consistent with accuracy ($\mathbf{C} > 0.5$), and AUC is statistically more discriminant than accuracy ($\mathbf{D} > 1$).

3.2.3 Comparison with Real-World Datasets

We conduct further experiments to calculate consistency and discriminancy of AUC and accuracy. But here we use real machine learning algorithms (instead of a simulation with enumeration of ranked lists) on real-world datasets (instead of artificial datasets). The algorithms we use are the standard C4.5 [51], Naive Bayes [18], and an improvement of C4.5 called C4.4 [48]. See Section 2.1 of Chapter 2 for detailed descriptions of these learning algorithms. Note that the actual selection of the algorithms are not an issue here, neither which one actually performs better in AUC or accuracy – such questions will be answered in Section 3.2.4. Here we only care if the two algorithms are consistent or not in AUC and accuracy, and if one measure is more discriminant than another. That is, we want to confirm Theorems 1 and 2 on real-world datasets using real learning algorithms.

We use 18 datasets (both binary and multi-class) with a relatively large number of examples from the UCI repository [4], as shown in Table 3.13. To confirm statistical consistency and discriminancy between accuracy and AUC on real-world datasets, 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 (Table 3.13). 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). Again, for each pair of algorithms, we do not care which one is better (this will be answered in Section 3.2.4.1); 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 3.14, and they are certainly consistent with the Theorems in Section 3.2.1. In the table's 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 3.14 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

Table 3.13: 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

84/(84+29)=0.743 is then the degree of consistency **C**. Similarly, the numbers in the row "acc=/AUC≠" indicates the number of cross-validation testing sets that the two algorithms are same in accuracy but different in AUC, and "acc≠/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 3.14, 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 version of C4.5, so it would produce similar results 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 3.16), and thus, they are more similar in the effect of AUC and accuracy on the testing datasets.

Table 3.14: 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

3.2.4 Comparing Learning Algorithms on AUC and Accuracy

We have established, empirically (Section 3.2.3) and formally (Section 3.2.1), that AUC is a better measure (using objective criteria of statistical consistency and discriminancy) than accuracy. Most previous work, however, 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 [51] and Naive Bayes are very similar in predictive accuracy [35, 36, 16]. How do popular learning algorithms, such as decision trees and Naive Bayes, compare in terms of the better measure AUC? How does the recent Support Vector Machine (SVM) method compare to traditional learning algorithms such as Naive Bayes and decision trees? We attempt to answer these questions in this section.

3.2.4.1 Comparing Naive Bayes and Decision Trees

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 the same 18 datasets (both binary and multi-class) with a relatively large number of examples from the UCI repository [4], as shown in Table 3.13. SVM is not involved in this comparison as some datasets are multiple classes (see Section 3.2.4.2 for details).

Our experiments follow the procedure below:

1. Continuous attributes in all datasets are discretized by the entropy-based method described in [21].

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. [1]

The averaged results on accuracy are shown in Table 3.15, and on AUC in Table 3.16. As we can see from Table 3.15, 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 [35, 36, 16].

Analyzing the results for AUC in Table 3.16, however, leads to an interesting conclusion. The 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 in [48]). That is, Naive Bayes outperforms C4.5 in AUC with significant difference.

This conclusion is quite significant for the machine learning and data mining community. Previous research concluded that Naive Bayes and C4.5 are very similar in

[1] Again the calculation of AUC depends only on the labeled examples (the true ranking is not needed) and learning algorithms which can produce probability estimations for ranking testing examples.

Table 3.15: 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**

prediction when compared by accuracy [35, 36, 16]. 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 favored over C4.5 in machine learning and data mining applications, especially when ranking is important.

3.2.4.2 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 [63, 14, 8] on the datasets from the UCI repository. Such an extensive comparison with a large number of benchmark datasets is still rare [44]; most previous works (such as Hastie etc. [27]) were limited to only a few comparisons, with the exception of Meyer etc. [44].

SVM is essentially a binary classifier, and although extensions have been made to multiclass classification [60, 28] there is no consensus which is the best. Therefore,

Table 3.16: 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**

we use the 13 binary-class datasets from the 18 datasets in the experiments involving SVM. Meyer etc. [44] also only used binary datasets for the classification for the same reason.

For SVM we use the software package LIBSVM [10] modified to directly output the evaluation of the distance from testing examples to the hyperplane with the maximal margin 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 [44]. The experiment procedure is the same as discussed in Section 3.2.3.

The predictive accuracy and AUC of SVM on the testing sets of the 13 binary datasets

are listed in Table 3.17. 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 3.15 we can obtain the average predictive accuracies of Naive Bayes, C4.4, and C4.5 on the 13 binary datasets are 85.9%, 86.5%, and 86.7%, respectively. Similarly, from Table 3.16 we can obtain the average predictive AUC values of Naive Bayes, C4.4, and C4.5 on the 13 binary datasets are 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 show 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 3.2.4.1). Note that in most previous comparisons, numerical attributes are used directly in SVM. In our experiments, however, we have discretized all numerical attributes (see Section 3.2.4.1) as Naive Bayes requires all attributes to be discrete. Discretization is an important pre-processing step in data mining [42]. 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. We believe that our comparisons are fair and valid 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. Also note that we did not seek problem-specific best kernels for SVM. This is fair as Naive Bayes, C4.5, and C4.4, are run automatically with the default, problem-independent parameter settings.

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

- The average predictive accuracies 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.

- 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 differ-

Table 3.17: 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**

ence than C4.5.

- AUC should replace accuracy in measuring and comparing classifiers as AUC is a better measure in general. This is particularly true as ranking is important in data mining applications, and AUC reflects ranking much more accurately and directly than accuracy.

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

3.2.5 Summary

In this section, we apply the formal definitions of discriminancy and consistency in comparing evaluation measures for learning algorithms. We establish precise and objective criteria for comparing two measures in general, and show, both empirically and formally, that AUC is a better measure than accuracy. This suggests that AUC should replace accuracy in measuring and comparing classifiers, as AUC is a better measure in general. We then reevaluate commonly accepted claims in machine learning based on accuracy using AUC, and obtain interesting and surprising new results.

We show that the average predictive AUC values of Naive Bayes, C4.4, and SVM are very similar (no statistical difference), and they are all higher than C4.5 with significant difference. This suggests that Naive Bayes, C4.4 and SVM should be preferred over C4.5 in real-world applications, especially when ranking is important.

The conclusions drawn in this section can have important implications in evaluating, comparing, and designing learning algorithms. In our future work, we will study the effect of data discretization on the performance of SVM and other algorithms, and we will redesign accuracy-based learning algorithms to optimize AUC. Some works have already been done in this direction.

3.3 Comparing Ranking Measures

In the previous research of the ranking issue in machine learning, a lot of work has focused on how to design or optimized ranking algorithms. As we have introduced in Chapter 1, some true ranking measures such as ED, SRN, MD and some partial ranking measures such as AUC are widely used in estimating how well a ranking is performed compared with the ideal ranking. In Statistics there are a lot of researches on the statistical properties of these ranking measures. However, little work has been done to directly compare these ranking measures. Just as the significance of comparing AUC and accuracy in the previous section, it is also important to perform a detailed and complete comparison among most commonly used rank measures.

In this section we compare six ranking measures of ED, MD, SRN, AUC, OAUC, accuracy (See Section 1.1.2) by using our general comparison criteria proposed in Section 3.1. By using artificial datasets we empirically study the degree of consistency and degree of discriminancy between every two ranking measures. Based on these results we obtain a preference order discovered for these measures (Section 3.3.1). We also perform experiments by using real-world datasets and ranking algorithms to confirm our preference order. It also shows that better ranking measures are more sensitive in comparing rank algorithms (see Section 3.3.2).

We first intuitively compare some pairs of measures and analyze whether any two measures satisfy the criteria of consistency and discriminancy. To begin with, we consider ED and MD because these two measures are quite similar in their definitions except that ED sums the squared distance while MD sums the absolute value. We

expect that these two measures are consistent in most cases. On the other hand, given a dataset with n examples there are a total of $O(n^3)$ different ED values and $O(n^2)$ different MD values. Thus ED is expected to be more discriminant than MD. Therefore we expect that ED is consistent with and more discriminant than MD.

For AUC and OAUC, since OAUC is an extension of AUC, intuitively we expect that they are consistent. Assuming there are n_1 negative examples and n_0 positive examples, the different values for OAUC is $n_1 \sum_{i=1}^{n_0} (n_1 + i)$, which is greater than the different values of AUC ($n_0 n_1$). We can also expect that OAUC is more discriminant and therefore better than AUC.

However for the rest of the ranking measures we cannot make these intuitive claims because they have totally different definitions or computational methods. Therefore, in order to perform an accurate and detailed comparison and to verify or overturn our intuitions, we will conduct experiments to compare all measures.

3.3.1 Comparing Ranking Measures on Artificial Datasets

To obtain the average degrees of consistency and discriminancy for all possible ranked lists, we use artificial datasets that consist of all possible ordered list of length 8 [2]. We assume that the ordered lists are uniformly distributed. We exhaustively compare all pairs of ordered lists and calculate the degree of consistency and degree of discriminancy between two rank measures for ordering.

Table 3.18 lists the degree of consistency between every pair of six rank measures for ordering. The number in each cell represents the degree of consistency between the measures in the same row and column of the cell. We can find that the degree of consistency between any two measures are greater than 0.5, which indicates that these measures are "similar" in the sense that they are more likely to be consistent than inconsistent.

Table 3.19 shows the degree of discriminancy among all 6 rank measures. The number in the cell of the ith row and the jth column is the degree of discriminancy for the measure in ith row over the one in jth column, and vice versa.

From these two tables we can draw the following conclusions. First, these results verified our previous intuitive conclusions about the relations between ED and MD,

[2]There are $n!$ different ordered lists for length n, so it is unfeasible to enumerate longer lists.

Table 3.18: Degree of consistency between pairs of ranking measures for ordering.

	AUC	SRN	MD	ED	OAUC	acc
AUC	1	0.88	0.89	0.87	0.99	0.98
SRN	0.88	1	0.95	0.98	0.89	0.91
MD	0.89	0.95	1	0.95	0.90	0.95
ED	0.87	0.98	0.95	1	0.88	0.90
OAUC	0.99	0.89	0.90	0.88	1	0.97
acc	0.98	0.91	0.95	0.90	0.97	1

Table 3.19: Degree of discriminancy between pairs of ranking measures for ordering.

	AUC	SRN	MD	ED	OAUC	acc
AUC	1	0.88	1.42	0.21	0.0732	14.0
SRN	1.14	1	1.84	0.242	0.215	9.94
MD	0.704	0.54	1	0.117	0.116	6.8
ED	4.76	4.13	8.55	1	0.87	38.2
OAUC	13.67	4.65	8.64	1.15	1	94.75
acc	0.071	0.10	0.147	0.026	0.011	1

and between AUC and OAUC. The degree of consistency between ED and MD is 0.95, and between AUC and OAUC 0.99, which means that ED and MD, and AUC and OAUC are highly consistent. The degree of discriminancy for ED over MD, and for OAUC over AUC are greater than 1, which means that ED is better than MD, and OAUC is better than AUC.

Second, since all values of the degree of consistency among all measures are greater than 0.5, we can decide which measure is better than another only based on the value of degree of discriminancy. Recall (Section 3.1) that a measure f is better than another measure g iff $\mathbf{C_{f,g}} > 0.5$ and $\mathbf{D_{f/g}} > 1$. The best measure should be the one whose degrees of discriminancy over all other measures are greater than 1. From Table 3.19 we can find that all the numbers in the OAUC row are greater than 1, which means that the measure OAUC's degrees of discriminancy over all other measures are greater than 1. Therefore OAUC is the best measure. In the same way we can find that ED is the second best measure, and SRN is the third best. AUC, MD, and acc are the worst.

Finally, we can obtain the following preference order for all six ranking measures:

$$OAUC \succ ED \succ SRN \succ AUC \succ MD \succ acc$$

From the preference order we can conclude that OAUC, a new measure designed based on AUC, is the best measure. ED is the close, second best. The difference for these two measures is not very large (the degree of discriminancy for OAUC over ED is only 1.15). Therefore, we should use OAUC and ED instead of others to evaluate ordering algorithms in most cases. Further, the two "partial" order classification measures AUC and accuracy do not perform well as compared with the true-order measures ED and SRN. This suggests that generally we should avoid using classification measures such as AUC and accuracy to evaluate ordering. Finally, MD is the worst true-order measure, and it is even worse than AUC. It should be avoided.

3.3.2 Comparing Ranking Measures with Ranking Algorithms

In this section, we perform experiments to compare two classification algorithms in terms of the six rank measures. What we hope to conclude is that the better rank measures (such as OAUC and ED) would be more sensitive to the significance test (such as the t-test) than other less discriminant measures (such as MD and accuracy). That is, OAUC and ED are more likely to tell the difference between two algorithms than MD and accuracy can. Note that here we do not care about which rank algorithm predicts better; we only care about the sensitivity of the rank measures that are used to compare the rank algorithms. The better the rank measure (according to our criteria), the more sensitive it would be in the comparison, and the more meaningful the conclusion would be for the comparison.

To generate true order of the lists in our experiments, the target attribute should be continuous, so the ground truth (i.e., the true order of the lists) can be established. Consequently, the classification algorithms should be able to accept continuous target for the training and testing, and to produce probability estimations to rank the testing examples. This way, we can compare the true order to the predicted order of the training and testing examples.

We choose Artificial Neural Networks (ANN) and Instance-Based Learning algorithm (IBL) as our algorithms as they can both accept and produce continuous target. The

ANN that we use has one hidden layer; the number of nodes in the hidden layer is half of the input layer (the number of attributes).

We use both artificial and real-world datasets to evaluate and compare ANN and IBL with the six rank measures. With artificial datasets we can easily control the size of the data, the number of attributes, the noise level, and the target function. We generate artificial datasets with 9 discrete attributes and 1 continuous target. We use the polynomial function $f(x) = ax^3 + bx^2 + cx + d$ to compute the continuous target, where x is the sum of all attribute values. Furthermore we randomly add noise to the target function. The target function is generated with the probability of 20% being a noise value. The training dataset contains 200 examples. We also select three real-world datasets *Wine*, *Auto-Mpg* and *CPU-Performance* from the UCI Machine Learning Repository [5]. The dataset *Auto-Mpg* has 9 attributes and 1 continuous class and contains 398 examples. The dataset *Wine* has 13 continuous attributes and 1 discrete class and contains 178 examples. For this dataset we exchange the first attribute with the discrete class to obtain a dataset with a continuous target. The dataset *CPU-Performance* has 6 continuous predictive attributes and 1 continuous target and contains 209 examples.

In our experiments, we run ANN and IBL with the 10-fold cross validation on the training datasets. For each round of the 10-fold cross validation we train the two algorithms on the same training data and test them on the same testing data. We measure the testing data with six different rank measures (OAUC, ED, SRN, AUC, MD and acc). We then perform paired, two-tailed t-tests on the 10 testing datasets for each measure to compare these two algorithms.

Table 3.20 shows the significance level in the t-test. [3] The smaller the values in the table, the more likely that the two algorithms (ANN and IBL) are significantly different, and the more sensitive the measure is when it is used to compare the two algorithms. Normally a threshold is set up and a binary conclusion (significantly different or not) is obtained. For example, if we set the threshold to be 0.95, then for the artificial dataset, we would conclude that ANN and IBL are statistically significantly different in terms of ED, OAUC and SRN, but not in terms of AUC, MD and acc. However, the actual significance level in Table 3.20 is more discriminant for the comparison. That is, it is "a better measure" than the simple binary classification

[3]The confidence level for the two arrays of data to be statistical difference is one minus the values in the table.

of being significantly different or not.

Table 3.20: The significance level in the paired t-test when comparing ANN and IBL using different rank measures.

Measures	artificial	Wine	Auto-mpg	CPU
OAUC	0.021	0.031	8.64×10^{-4}	1.48×10^{-3}
ED	0.011	0.024	1.55×10^{-3}	4.01×10^{-3}
SRN	0.039	0.053	8.89×10^{-3}	5.91×10^{-3}
AUC	0.084	0.062	5.77×10^{-3}	8.05×10^{-3}
MD	0.116	0.053	0.0167	5.97×10^{-3}
acc	0.239	0.126	0.0399	0.0269

From Table 3.20 we can obtain the preference order from the most sensitive measure (the smallest significance level) to the least sensitive measure (the largest significance level) for each dataset is:

- Artificial: ED, OAUC, SRN, AUC, MD, acc.
- Wine: ED, OAUC, SRN = MD, AUC, acc.
- Auto-mpg: OAUC, ED, AUC, SRN, MD, acc.
- CPU-Performance: OAUC, ED, SRN, MD, AUC, acc.

These preference orders are roughly the same as the preference order of these measures discovered in the last section:

$$OAUC \succ ED \succ SRN \succ AUC \succ MD \succ acc$$

There are several cases where ED and OAUC, AUC and MD, AUC and SRN are switched in their order. As we have seen earlier, the difference on discriminancy between ED and OAUC, AUC and MD, AUC and SRN are small ($D_{OAUC/ED} = 1.15$, $D_{AUC/MD} = 1.42$, $D_{SRN/AUC} = 1.14$). In any case, the sensitivity order of the rank measures may be slightly different for individual datasets.

The experimental results confirm our analysis in the last section. That is, OAUC and ED are the best rank measures for evaluating orders. In addition, MD and accuracy should be avoided as rank measures. These conclusions will be very useful

for comparing and constructing machine learning algorithms for ranking, and for applications such as Internet search engines and data mining for CRM (Customer Relationship Management).

3.3.3 Summary

In many real-world applications, such as information retrieval and CRM (Customer Relationship Management) in data mining, accurate ranking is crucial. Many rank measures have been used but little theoretical work and practical guideline have been given to compare them. In this section, we propose a new rank measure (OAUC) for ordering, and compare it together with five commonly used rank measures for ordering. We conclude that OAUC is actually the best ranking measure, and it is closely followed by the Euclidean distance (ED). Our results indicate that in comparing different algorithms for the order performance, we should use OAUC or ED, and avoid the least sensitive measures such as Manhattan distance (MD) and accuracy.

Chapter 4

Constructing New and Better Machine Learning Measures

In Section 3.1 we set up criteria for comparing two evaluation measures and for determining if a measure is better than another. In this chapter we describe general methods to construct new measures from existing one, and we prove that the new measures are better (according to the criteria established in Section 3.1) than the existing ones. This is very useful in real-world data mining applications as better measures should always be used in comparing different learning algorithms. In addition, as we will show in Section 4.3, learning algorithms optimized with better measures also predict better.

4.1 Construction Approaches

One might think that the simplest method to construct a new and better measure is to build upon a single existing measure. For example, given accuracy as an existing measure, can 10 times accuracy (10 $\times$ accuracy) be a better measure? Obviously not (as they are equivalent, or there is a one-to-one mapping between these two measures). The following theorem shows that in general it is not possible to build a new and better measure based on a monotonic transformation (such as a linear function) of a single measure.

Theorem 3 For any measure f, assume that $\phi(f)$ is a function of f representing a

new measure. If ϕ is monotonically incremental[1], then $\phi(f)$ is equivalent to f.

Proof: Since $\phi(f)$ is monotonically incremental, then for any two objects a, b, $f(a) > f(b)$ iff $\phi(f(a)) > \phi(f(b))$. Therefore, $\phi(f)$ is equivalent to f, according to Definition 1 in Section 3.1. □

Theorem 3 shows that we can only possibly construct a new measure from two or more measures.

In the rest of this section, we propose two approaches to construct new measures based on two existing ones, and prove that the new measures are better than the existing ones. We will use AUC and accuracy as the two existing measures in experimental verification.

4.1.1 Two-level Measures

Our first approach is to construct a "two-level measure", denoted as $f : g$, based on two measures f and g. Our intuitive idea of the two-level measure comes from sports. In certain soccer tournaments, many teams compete for a champion. The team who wins the largest number of the games will be the championship. However, if two teams win the same number of games, the team with the higher total number of goals earns the championship. The number of winning games is the first measure, and the total number of goals in the games is another measure, secondary to the first one.

We can define the two-level measure $f : g$ as follows.

Definition 11 A two-level measure ϕ formed by f and g, denoted by $f : g$, is defined as: $\phi(a) > \phi(b)$ iff $f(a) > f(b)$, or $f(a) = f(b)$ and $g(a) > g(b)$; and $\phi(a) = \phi(b)$ iff $f(a) = f(b)$ and $g(a) = g(b)$.

The following is a simple approximation to form a two-level measure from AUC and accuracy. If we keep 3 digits for the values of AUC and accuracy, then a new two-level measure can be $1000 \times AUC + acc$. For example, when $AUC = 0.567$ and $acc = 0.123$, the two-level measure is thus $1000 \times 0.567 + 0.123 = 567.123$. Clearly, this new measure satisfies the two-level measure definition (if AUC and accuracy are

[1]For any a and b, $a < b$ iff $f(a) < f(b)$. Clearly this theorem also applies to monotonically decremental functions.

only accurate up to the 4th decimal place). That is, when comparing two learning algorithms, if AUC of the first algorithm is larger, the new two-level measure of the first algorithm is always larger no matter what the value of accuracy is; but if AUC is the same, then the larger the accuracy, the larger the two-level measure – similar to the example of the the soccer tournaments given above.

The following theorem shows the degree of consistency and the degree of discriminancy between the two-level measure defined earlier and the existing measures.

Theorem 4 Let $\phi = f : g$ be the two-level measure formed by f and g, $f \succ g$, and $\mathbf{D}_{f/g} \neq \infty$. Then $\mathbf{C}_{\phi,f} = 1$, and $\mathbf{D}_{\phi/f} = \infty$. In addition, $\mathbf{C}_{\phi,g} \geq \mathbf{C}_{f,g}$, and $\mathbf{D}_{\phi/g} = \infty$. That is, ϕ is a better measure than both f and g; i.e., $\phi \succ f \succ g$.

Proof:[2] By Definition 11 there does not exist objects a, b such that "$f : g(a) > f : g(b), f(a) < f(b)$". Therefore $INCON_{\phi,f} = 0$, $\mathbf{C}_{\phi,f} = 1$. Since for any a, b such that "$f : g(a) > f : g(b), g(a) > g(b)$" is equivalent to "$f(a) = f(b), g(a) > g(b)$" and "$f(a) > f(b), g(a) > g(b)$", we have $CON_{\phi,g} = CON_{f,g} + DIS_{g/f}$. "$f : g(a) > f : g(b), g(a) < g(b)$" is equivalent to "$f(a) > f(b), g(a) < g(b)$", thus $INCON_{\phi,g} = INCON_{f,g}$. Therefore $\mathbf{C}_{\phi,g} \geq \mathbf{C}_{f,g} > 0.5$. For discriminancy there does not exist a, b such that "$f : g(a) = f : g(b)$ and $f(a) > f(b)$". Since $\mathbf{D}_{f/g} > 1$, $\mathbf{D}_{f/g} \neq \infty$, there exists $a, b \in \Psi$ such that "$f(a) = f(b)$ and $g(a) > g(b)$" which is equivalent to "$f : g(a) \neq f : g(b)$ and $f(a) = f(b)$". Therefore $\mathbf{D}_{\phi/f} = \infty$, similarly we have $\mathbf{D}_{\phi/g} = \infty$. □

Theorem 4 indicates that the two-level measure $f : g$ is indeed a better measure than f and g. Further, we can prove that $f : g$ dominates f. That is, $f : g$ to f is analogous to numerical marks to letter marks – there are no cases that the two measures would disagree.

Theorem 5 $f : g$ dominates f.

Proof: By Definition 11 there does not exist a, b such that $f : g(a) > f : g(b)$ and $f(a) < f(b)$. Therefore for any a, b, $f : g(a) > f : g(b)$ implies $f(a) \geq f(b)$. □

To confirm Theorem 4 when it applies to the two-level measure $AUC : acc$, we conduct experiment to compute the five percentage criteria (described in Section 3.1) between

[2]We must show both $\phi \succ f$ and $\phi \succ g$ because the relation $\succ$ is not transitive in general.

the AUC : acc, AUC, and acc. This also gives us an intuition on the degree of the consistency, inconsistency, discriminancy, and indifference between AUC : acc, AUC, and acc.

To conduct the experiment, we exhaustively enumerate all possible pairs of ranked lists with 6, 8, 10, 12, 14, and 16 examples of artificial datasets with an equal number of positive and negative examples. The five percentage criteria are computed, and the results are shown in Tables 4.1. We also draw two pie charts showing the percentages of all five criteria between AUC : acc, AUC, and acc in in Figures 4.1 and 4.2 respectively. Clearly, we can see from the tables and figures that $\mathbf{C}_{\phi,\mathbf{AUC}} = 1$, and $\mathbf{D}_{\phi/\mathbf{AUC}} = \infty$. Similarly, we can see that $\mathbf{C}_{\phi,\mathbf{acc}} > \mathbf{C}_{\mathbf{AUC},\mathbf{acc}}$, and $\mathbf{D}_{\phi/\mathbf{acc}} = \infty$. These are consistent with Theorem 4.

Another conclusion we can draw about AUC : acc is that from Figure 4.1 we can see that the CON (97%) is very close to 1, while others are very small or zero. This indicates that the two level measure AUC : acc is highly consistent with AUC. On the other hand, the consistency between AUC : acc and acc is much lower (but still greater than 0.5). That is, AUC : acc is much more consistent with AUC than with acc.

Table 4.1: Compare the two-level measure ϕ=AUC : acc with AUC.

#	CON	INCON	DIS(ϕ/A)	DIS(A/ϕ)	IND	C	$D_{\phi/A}$
6	0.926	0	0.021	0	0.053	1	∞
8	0.934	0	0.022	0	0.045	1	∞
10	0.946	0	0.020	0	0.034	1	∞
12	0.957	0	0.017	0	0.026	1	∞
14	0.964	0	0.015	0	0.020	1	∞
16	0.970	0	0.014	0	0.016	1	∞

One might think that we could construct an even better "three-level" measure (such as $(f : g) : f$) from the newly formed two-level measure $f : g$ and an original measure f or g, and this process could repeat to get better and better measures. However, this will not work. Recall that in Theorem 4 one of the conditions to construct a better two-level measure $\phi = f : g$ is that $\mathbf{D}_{f/g} \neq \infty$. However, Theorem 4 proves that $\mathbf{D}_{\phi/f} = \mathbf{D}_{\phi/g} = \infty$, making it impossible for ϕ to be combined with f or g for further constructing new measures. Therefore, we can use this method of constructing a

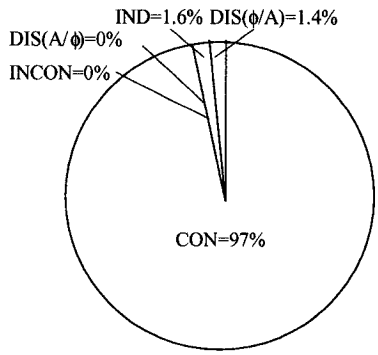


Figure 4.1: Illustrations of the five percentage criteria between ϕ=AUC : acc with AUC.

Table 4.2: Compare the two-level measure ϕ=AUC : acc with acc.

#	CON	INCON	DIS(ϕ/a)	DIS(a/ϕ)	IND	C	$D_{\phi/a}$
6	0.616	0.005	0.326	0	0.053	0.992	∞
8	0.626	0.014	0.316	0	0.045	0.978	∞
10	0.644	0.024	0.298	0	0.034	0.964	∞
12	0.659	0.033	0.282	0	0.026	0.953	∞
14	0.671	0.040	0.268	0	0.020	0.943	∞
16	0.683	0.046	0.255	0	0.016	0.936	∞

two-level measure from two existing measures only *once*.

4.1.2 Linear Combinations

The second approach for constructing a new measure is linear combination $\alpha f + \beta g$ from two measures f and g. Without loss of generality, we assume $\alpha > 0$ and $\beta > 0$, and we normalize α and β such that $\alpha + \beta = 1$. Thus the new measure $\phi(f, g) = \alpha f + (1 - \alpha)g$. We use $f \oplus g$ to represent this measure. The following theorem proves that the consistency between the new measure $f \oplus g$ and f (or g) is more than that between f and g.

Theorem 6 Let $\phi(f, g) = f \oplus g = \alpha f + (1-\alpha)g$. Then $\mathbf{C_{\phi,f}} \geq \mathbf{C_{f,g}}$, and $\mathbf{C_{\phi,g}} \geq \mathbf{C_{f,g}}$.

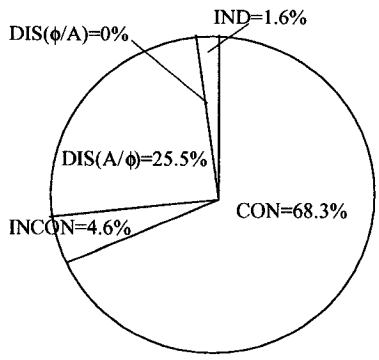


Figure 4.2: Illustrations of the five percentage criteria between ϕ=AUC : acc with acc.

Proof: Let $R = \{(a,b)|f(a) > f(b), g(a) > g(b)\}$, and $S = \{(a,b)|f(a) > f(b), g(a) < g(b)\}$. $R' = \{(a,b)|\phi(a) > \phi(b), f(a) > f(b)\}$, and $S' = \{(a,b)|\phi(a) < \phi(b), f(a) > f(b)\}$. Since $\phi = \alpha f + (1-\alpha)g$, it is obvious that $R' = \{(a,b)|\phi(a) > \phi(b), f(a) > f(b)\} \supseteq R + \{(a,b)|f(a) > f(b), g(a) = g(b)\}$, and $S' \subseteq S$. Therefore, $|R'| \geq |R|$, and $|S'| \leq |S|$. Since $\mathbf{C_{f,g}} = \frac{|R|}{|R|+|S|}$, thus $\mathbf{C_{\phi,f}} = \frac{|R'|}{|R'|+|S'|} \geq \mathbf{C_{f,g}}$. Similarly we can prove that $\mathbf{C_{\phi,g}} \geq \mathbf{C_{f,g}}$. □

The following theorem shows that under certain conditions, the new linearly combined measure $f \oplus g$ is also more discriminant than f (or g).

Theorem 7 Let f and g be any two measures represented by rational numbers, and α be an irrational number. Let $\phi(f,g) = f \oplus g = \alpha f + (1-\alpha)g$. If $\mathbf{D_{f/g}} \neq \infty$, $\mathbf{D_{g/f}} \neq \infty$, then $\mathbf{D_{\phi/f}} = \infty$, and $\mathbf{D_{\phi/g}} = \infty$.

Proof: Let $R = \{(a,b)|\phi(f,g)(a) = \phi(f,g)(b), f(a) \neq f(b)\}$, $S = \{(a,b)|\phi(f,g)(a) \neq \phi(f,g)(b), f(a) = f(b)\}$. Then $\mathbf{D_{\phi/f}} = \frac{|S|}{|R|}$. $R = \{(a,b)|\alpha f(a) + (1-\alpha)g(a) = \alpha f(b) + (1-\alpha)g(b), f(a) \neq f(b)\} = \{(a,b)|f(a) - f(b) = \frac{(1-\alpha)}{\alpha}(g(b) - g(a)), f(a) \neq f(b)\}$. As $\frac{(1-\alpha)}{\alpha}$ is an irrational number, it is impossible that $f(a) - f(b) = \frac{(1-\alpha)}{\alpha}(g(b) - g(a))$ when $f(a) \neq f(b)$. Therefore $R = \Phi$. Since $\mathbf{D_{f/g}} \neq \infty$, $S \neq \Phi$, so we have $\mathbf{D_{\phi/f}} = \infty$. Similarly we can prove $\mathbf{D_{\phi/g}} = \infty$. □

Similarly, to test Theorems 6 and 7, we conduct experiments that apply to AUC and acc. We conduct the same experiments on artificial datasets as we did previously. When choosing $\alpha = \frac{\sqrt{2}}{2}$, the experimental results are shown in Tables 4.3 and 4.4. Clearly, the results confirm Theorems 6 and 7.

Table 4.3: Comparing $AUC \oplus acc = \alpha AUC + (1-\alpha)acc$ with AUC in terms of five percentage criteria.

#	CON	INCON	DIS(ϕ/A)	DIS(A/ϕ)	IND	**C**	$D_{\phi/A}$
6	0.926	0	0.021	0	0.053	1	∞
8	0.934	0	0.022	0	0.045	1	∞
10	0.946	0	0.020	0	0.034	1	∞
12	0.957	0	0.017	0	0.026	1	∞
14	0.965	0	0.015	0	0.020	0.99	∞
16	0.970	0	0.014	0	0.016	0.99	∞

Table 4.4: Comparing $AUC \oplus acc = \alpha AUC + (1-\alpha)acc$ with acc in terms of five percentage criteria.

#	CON	INCON	DIS(ϕ/a)	DIS(a/ϕ)	IND	**C**	$D_{\phi/a}$
6	0.616	0.005	0.326	0	0.053	0.992	∞
8	0.626	0.014	0.316	0	0.045	0.978	∞
10	0.644	0.024	0.298	0	0.034	0.964	∞
12	0.659	0.033	0.282	0	0.026	0.953	∞
14	0.672	0.040	0.268	0	0.020	0.944	∞
16	0.683	0.046	0.256	0	0.016	0.937	∞

Similar to the two-level construction that can be used only once to construct a new measure, it is easy to show that the linear combination can also be used only once to create a better measure.

What about the two newly constructed measures, $AUC : acc$ and $AUC \oplus acc$? Which one is better? Using the same five percentage criteria, we can also compare the two newly constructed measures. By comparing the results between Table 4.1 and 4.3, and between Table 4.2 and 4.4, we can see that the corresponding five percentage criteria are approximately same. Thus we can conclude that the new two-level measure and the linear combination of two measures have roughly the same performance as measures for the learning algorithms.

The results above also show advantages of the new five percentage criteria proposed in this thesis. If we just look at **C** and **D** between $AUC : acc$ and AUC (as in Table 4.1), we might easily conclude that $AUC : acc$ is much better than AUC, as

$\mathbf{C} = 1$ and $\mathbf{D} = \infty$. The five percentage criteria tell us that $AUC : acc$ and AUC are in fact very similar, as the consistency (CON) of the two is very large (over 0.9). Similarly, it is the five percentage criteria that indicate that the two newly constructed measures $AUC : acc$ and $AUC \oplus acc$ are very similar. Thus, the newly proposed, more refined five percentage measures have greater advantages over the previously proposed measures [37, 29].

These general methods of constructing new measures are very useful in evaluating learning algorithms. For example, we have already shown that AUC is a better measure than accuracy, and therefore, AUC should replace accuracy in comparing two learning algorithms. But the two-level measure formed from AUC and accuracy goes one step further. That is, when comparing two learning algorithms, if AUC is the same on a testing set, then we compare the accuracy to see which one is better. This gives rise to a more discriminant evaluation in comparing learning algorithms than using AUC alone. Again the important feature of our methods is that they are general methods for constructing better measures based on two existing ones, regardless of the domains and problems we are working on.

Another advantage of discovering better measures is that learning algorithms should always try to build a model by optimizing better measures. We will discuss this in Section 4.3.

4.2 Comparing to RMS

In the previous sections, we have shown methods of constructing new measures (the two level measure and linear combination), and proved formally that the new measures are better than the existing ones. The comparison between the new and existing ones is based on the five percentage criteria. As we indicated earlier, these criteria are "internal"; that is, they are used to compare the relative differences between two arbitrary measures. They may not reflect how close these measures would be compared to a true performance measure in real-world applications.

In this section we compare the newly constructed measures and the existing ones to an "ideal" measure, and we hope to show that the better the measure, the more close it is to the ideal measure. That is, we hope to show that the newly constructed measures are not only better with respect to internal criteria, but also better with respect to an external criterion.

Clearly, the true measure in different applications varies. In many applications, it is often difficult to obtain the true measure before models are learned from data. Therefore, we often choose the most "strict" measure in optimization (model building). RMS (Root Mean Square) error [34] is often chosen to optimize in various applications (such as in management science [54], Economics [11] and Bioinformatics [3]), and we also choose RMS as the ideal measure to which the newly constructed and existing measures are compared.

It is easy to see why RMS is suitable for this job. All algorithms for ranking not only predict the class, but also probabilities of the classes. Given a set of examples with the true probabilities for all classes, the ideal ranking algorithm would predict the exact same probabilities as the true ones. In this ideal case, the RMS is 0. If the predicted probabilities are within a small perturbation of the true probabilities, the RMS will be small and will reflect the size of perturbation, even if the rank may not be altered. If the perturbation is large, the rank will be altered and the RMS will be too large to reflect that. Thus, RMS is the "most strict" measure for ranking and classification.

To experimentally compare RMS with the newly constructed and existing measures, we use AUC and accuracy again. In the last section, we construct the two-level measure $AUC : acc$ and the linear combination $AUC \oplus acc$ based on AUC and acc, and show that $AUC : acc$ and $AUC \oplus acc$ are very similar, and they are both better than AUC, which is in turn, better than acc. Now we want to verify experimentally that $AUC : acc$ and $AUC \oplus acc$ are more correlated with RMS than AUC and acc.

We first randomly generate pairs of "true" ranked lists and perturbated ranked lists. The "true" ranked list always consists of n binary examples, with the i-th example having the probability of $p_i = \frac{i}{n}$ of belonging to the positive class. We then generate a perturbed ranked list by randomly fluctuating the probability of each example within a range bounded by ϵ. That is, if the true probability is p, the perturbed probability is randomly distributed in $[max(0, p_i - \epsilon), min(1, p_i + \epsilon)]$. Table 4.5 shows an example of the "true" and perturbed ranked lists with 10 examples. Examples with the probability of greater than 0.5 are regarded as positive, and otherwise as negative. From this table the values of RMS, AUC, acc, $AUC : acc$ and $AUC \oplus acc$ compared to the "true" ranked list can be easily computed as 0.293, 0.68, 0.6, 0.686 and 0.657 respectively.

After we generate many tuples of RMS, AUC, acc, $AUC : acc$ and $AUC \oplus acc$, we can

Table 4.5: An example of "true" and perturbated ranked lists.

True	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
	−	−	−	−	−	+	+	+	+	+
Perturbated	0.0	0.15	0.6	0.5	0.95	0.2	0.65	0.7	1.0	0.4
	−	−	+	−	+	−	+	+	+	−

calculate correlation coefficients of AUC, acc, $AUC : acc$ and $AUC \oplus acc$ compared to RMS[3]. The correlation coefficient is defined as follows. For two series values of a_i and b_i, suppose that the average values of these two series values are $\bar{a}$ and $\bar{b}$ respectively. Let $S_{aa} = \sum(a_i - \bar{a})^2$, $S_{bb} = \sum(b_i - \bar{b})^2$, $S_{ab} = \sum(a_i - \bar{a})(b_i - \bar{b})$. Then the correlation coefficient is $r = \frac{S_{ab}}{\sqrt{S_{aa}S_{bb}}}$. Clearly, the correlation coefficient measures how well two series of values are correlated. If $r = 0$, it indicates that the two series of values are random. If $r = 1$, the two series of values are perfectly positively correlated. If $r = -1$, the two series are perfectly negatively correlated. In general, the larger the $|r|$, the stronger the correlation between two series of values.

We calculate the correlation coefficients between RMS and AUC, acc, $AUC : acc$, and $AUC \oplus acc$ respectively, after generating 200 tuples of these values. For each, we also vary the value of ϵ. We repeat this process five times, and the averaged correlation coefficients are listed in Table 4.6. We perform a two-tailed paired t-test with 95% confidence interval to see whether the differences in these correlation coefficients are statistically significant. The values in bold means that they are significantly indifferent with the largest values in each row, but they are significantly larger than the values not in bold.

Several interesting conclusions can be drawn from the table. First of all, the correlation coefficients of AUC, $AUC : acc$ and $AUC \oplus acc$ are all significantly greater than that of acc with all ϵ values. It indicates that these measures are all better than accuracy as they are "closer" to RMS. This confirms our early conclusion that these measures are better than accuracy according to our five percentage criteria.

Secondly, we can see that when ϵ is small (0.3 and 0.5), there is no significant difference in correlation coefficients of AUC, $AUC : acc$, $AUC \oplus acc$. But with the increasing value of ϵ, our newly constructed measures $AUC : acc$ and and $AUC \oplus acc$

[3]Actually AUC, acc, $AUC : acc$ and $AUC \oplus acc$ are all compared to $(1 - RMS)$ as it correlates positively with other measures: the larger, the better.

Table 4.6: Comparing correlation coefficients of *acc*, *AUC*, *AUC* : *acc*, and *AUC*⊕*acc* with *RMS*.

	acc	AUC	AUC⊕acc	AUC:acc
$\epsilon = 0.3$	0.2454±0.076	**0.3222±0.072**	**0.3024±0.073**	**0.3177±0.072**
$\epsilon = 0.5$	0.4678±0.071	**0.535±0.063**	**0.5297±0.067**	**0.5352±0.064**
$\epsilon = 0.7$	0.5932±0.01	**0.6596±0.015**	**0.660±0.013**	**0.6616±0.014**
$\epsilon = 0.8$	0.6546±0.051	0.6996±0.041	**0.7067±0.041**	**0.7036±0.041**
$\epsilon = 0.9$	0.6656±0.024	0.7137±0.025	**0.7188±0.025**	**0.7168±0.025**

become significantly more correlated with RMS than *AUC*. This again verifies our early conclusion that the newly constructed measures are better according to our five percentage criteria.

Thirdly, when ϵ is small (0.3), the values of all correlation coefficients are relatively small (from 0.2454 to 0.3177), but when ϵ is large (0.9), the values are larger (from 0.6656 to 0.7188). This can be understood, as when the perturbation (ϵ) is small, there can often be no change in ranking ($AUC = 1$) and accuracy ($acc = 1$). Thus the values of *AUC* and *acc* do not correlate well with RMS. When the perturbation (ϵ) is large, the rank list (AUC) and accuracy are both affected.

Lastly, there is no significant difference between correlation coefficients of *AUC* : *acc* and *AUC*⊕*acc*. This also verifies early conclusion that the difference between *AUC* : *acc* and $AUC \oplus acc$ is very small according to our five percentage criteria.

In sum, from the experiments conducted in this section, we can conclude that *AUC* : *acc* and $AUC \oplus acc$ have almost the same correlation with RMS, and both are more correlated than *AUC*, which is more correlated than accuracy. We can see that this confirms our early conclusion that $AUC : acc = AUC \oplus acc \succ AUC \succ acc$.

In the next section, we will perform experiments to compare the performance of different Artificial Neural Networks optimized with the measures of *AUC* : *acc*, *AUC*⊕*acc*, *AUC*, *acc* respectively.

4.3 Building Models with Better Measures

In Section 4.1, we showed that the two-level measure $AUC : acc$ is better than AUC (which is in turn better than accuracy). That is, $AUC : acc \succ AUC \succ acc$. As we have also discussed earlier, a significant advantage of discovering better measures is that they can be used in building learning models (such as classifiers) by optimizing the better measures directly. For example, most decision trees are built by minimizing the entropy, or maximizing the accuracy. In this section, we will show that by maximizing $AUC : acc$ or AUC, we will get better learning models than by maximizing the accuracy.

We will conduct our experiments using artificial neural networks (ANNs). This is because ANNs are much more sensitive to small changes in the optimization process to produce different weights that are continuous numbers. On the other hand, decision trees may not be sensitive enough to changes in the attribute selection criterion. For example, using AUC and accuracy, the same or very similar attributes could be selected in the tree building process. [22] showed that building decision trees with AUC did not lead to a significant improvement measured in AUC or accuracy.

Essentially we want to train three ANNs with the same training data optimized using $AUC : acc$, AUC, and acc respectively. For simplicity, we call the three ANN models $\text{ANN}_{AUC:acc}$, ANN_{AUC}, and ANN_{acc} respectively. Then we test these three ANN models on the testing tests. The predictive performance of the three different learning models on the test sets are measured by $AUC : acc$, AUC, and acc. We do this many times (using a 10-fold cross-validation) to obtain the averages on testing $AUC : acc$, AUC, and accuracy. What we hope to see is that the model optimized by $AUC : acc$ predicts better than the model optimized by AUC, measured by all of the three measures ($AUC : acc$, AUC, and acc). Similarly, the model optimized by AUC would be better than the model optimized by accuracy.

To optimize ANN with a measure f (f is either $AUC : acc$, AUC, or acc here), we implement the following simple optimization algorithm for neural networks. We still use the standard back-propagation algorithm that minimizes the sum of the squared differences (same as the RMS error used in Section 4.2, as it is the most "strict" measure), but we monitor the change in f instead to decide when to stop training. More specifically, we save the current weights in the neural network, and look ahead and train the network for N epochs, and obtain the new f value. If the difference

between the two f values is larger than a pre-selected threshold ϵ, it indicates that the neural network is still improving according to f, so we save the new weights (after training N epochs) as the current best weights, and the process repeats. If the difference between the two f values is less than ϵ, it indicates that the neural network is not improving according to f, so the training stops, and the saved weights are used as the final weights for the neural network optimized by f.

We choose $\epsilon = 0.01$ and $N = 50$. We choose 8 real-world datasets from the UCI Machine Learning Repository [5]. Each dataset is split into training and test sets using 10-fold cross-validation. The predictive performance on the testing sets from the three models $\text{ANN}_{AUC:acc}$, ANN_{AUC}, and ANN_{acc} is shown in Table 4.7. We perform a two-tailed paired t-test with 95% confidence interval on the averaged values of acc, AUC, and $AUC : acc$ predicted by three models. In Table 4.7 we mark the average value with a "*" to indicate that it is significantly better (larger) than the value immediately below it.

Several very interesting and surprising conclusions can be drawn from the averaged results (in bold) in Table 4.7. First of all, measured by accuracy, we can see that the ANN model optimized by $AUC : acc$ has the highest average accuracy (0.8052), than the ANN model optimized by AUC (the average accuracy is 0.7907), than the ANN model optimized by accuracy (the average accuracy is 0.7811), although the differences are not statistically significant. This is somewhat against a common intuition in machine learning that a model should be optimized by a measure that it will be measured on. In general this intuition is true. If your child will be tested on math, it is best to train her on math instead of French. However, if you have a better (consistent and more discriminant) measure, it is better off to train the model using the better measure, even if the result will be evaluated by the original measure. As a crude analogy, if the math test is graded by the letter marks, it would be more advantageous to improve your child's math skills by training her with the numerical marks. This way, she will be optimizing her scores in numerical marks (with finer improvements). After she does so, her test score (still measured by the letter marks) would be better than or equal to the score if she was trained and optimized by the letter marks.

Secondly, we can compare the averaged predictive results vertically[4], and draw the

[4]It is not meaningful to compare results in Table 4.7 horizontally as values of accuracy, AUC, and $AUC : acc$ are not comparable.

same conclusions. We can see that, measured by AUC, the ANN model optimized by $AUC : acc$ performs the best, then the model optimized by AUC, then the model optimized by accuracy. The same conclusion can be drawn for the measure $AUC : acc$. This shows the advantage of using better measures in model building – optimizing better measures lead to models with better predictions.

Note that our approach of improving predictive performance of machine learning and data mining algorithms is general as previous methods based on accuracy optimization can be re-designed and improved by optimizing better measures discovered.

One might wonder if we should simply use RMS for optimizing all learning models. If the true probabilities are known, then results of this chapter indicate that indeed RMS should be used to compare and optimize models. On the other hand, only class labels are given in many real-world datasets. AUC and accuracy (or $AUC : acc$) rely only on classification labels. The results of this chapter indicate that $AUC : acc$ (or AUC) should replace accuracy in comparing and optimizing models in these cases.

4.4 Summary

Evaluation metrics are essential in machine learning and other experimental science and engineering areas. In Chapter 3, we established general criteria to compare the predictive performance of any two single-number measures. We propose general approaches to construct new measures based on existing ones, and prove that the new measures are better than the existing ones according to the proposed criteria. We then compare experimentally the new measures with an ideal measure, and show that it is more correlated with it than the existing measures. Finally, we show that learning models optimized by the new and better measure predict better than models optimized by the existing ones. Our work is significant because not only can better measures compare and distinguish more accurately between learning algorithms, the predictive performance of learning algorithms can also be improved when better measures are optimized in building learning models.

Table 4.7: Predictive results from the three ANNs optimized by $AUC : acc$, AUC, and accuracy. The average value with a "*" indicates that it is significantly better (larger) than the value immediately below it.

Dataset	Model	acc	AUC	$AUC : acc$
australia	$ANN_{AUC:acc}$	0.7058	0.6518	0.6589
	ANN_{AUC}	0.6936	0.6245	0.6314
	ANN_{acc}	0.7058	0.652	0.659
breast	$ANN_{AUC:acc}$	0.8432	0.6531	0.6615
	ANN_{AUC}	0.8446	0.6553	0.6637
	ANN_{acc}	0.8447	0.6527	0.6611
cars	$ANN_{AUC:acc}$	0.8686	0.9197	0.9284
	ANN_{AUC}	0.8643	0.9297	0.9383
	ANN_{acc}	0.7829	0.7248	0.7326
eco	$ANN_{AUC:acc}$	0.9488	0.934	0.9435
	ANN_{AUC}	0.8497	0.9436	0.9521
	ANN_{acc}	0.9548	0.9458	0.9553
heart	$ANN_{AUC:acc}$	0.7778	0.8101	0.8178
	ANN_{AUC}	0.7854	0.8163	0.8242
	ANN_{acc}	0.7778	0.8098	0.8176
hepatitis	$ANN_{AUC:acc}$	0.8305	0.805	0.8133
	ANN_{AUC}	0.8305	0.805	0.8133
	ANN_{acc}	0.8305	0.6503	0.6586
pima	$ANN_{AUC:acc}$	0.7041	0.7208	0.7279
	ANN_{AUC}	0.699	0.7129	0.7199
	ANN_{acc}	0.6068	0.5488	0.5549
voting	$ANN_{AUC:acc}$	0.7627	0.6802	0.6878
	ANN_{AUC}	0.7586	0.6588	0.6664
	ANN_{acc}	0.7456	0.6324	0.6399
Average	$ANN_{AUC:acc}$	**0.8052**	**0.7718**	**0.7799**
	ANN_{AUC}	**0.7907**	**0.7683***	**0.7762***
	ANN_{acc}	**0.7811**	**0.7021**	**0.7099**

Chapter 5

Model Selection with Measures

Model selection is a significant task in machine learning and data mining. Among a set of models, it attempts to select the model that neither underfits nor overfits for future unseen data. Since performance measures can be used to evaluate the performance of learning models, they can be used to do model selection. However, it is still not clear how different measures perform in model selection. In this chapter we thoroughly explore the model selection abilities of nine measures under highly uncertain situations. We show that generally we should not use the goal measure (see Section 5.1) to do model selection. We also show that a measure's model selection ability is stable to class distributions and model selection goals, while different learning algorithms should choose different measures in model selection.

5.1 Model Selection Under Highly Uncertain Situations

Some machine learning and data mining tasks, such as facial and hand writing recognitions, usually need to train a highly robust and accurate learning model. In these cases a learning model trained with the default or arbitrary parameter settings is not enough because it usually cannot achieve the best performance. To satisfy these requirements we vary the parameter settings to train more than one learning model and then select the best one as the desired model. Instances of selecting learning model include choosing the optimal number of hidden nodes in neural networks, choosing the

optimal parameter settings of Support Vector Machines, and determining the suitable amount of pruning in building decision trees. This gives rise to the *model selection* problem, which is an important task in statistical estimation, machine learning, and scientific inquiry [62, 41]. *Model selection* attempts to select the model with best future performance from alternate models measured with a model selection criterion. Traditional model selection tasks usually use accuracy as model selection criterion. However, some data mining applications often call for other measures as criteria. For example, ranking is an important task in machine learning. If we want to select a model with best future ranking performance, then AUC (Area Under the ROC Curve), instead of accuracy, should be used as the model selection criterion. A model selection criterion is called a "model selection goal". Holdout testing method is a primary approach to perform model selection. It uses holdout data to estimate a model's future performance: repeatedly using a subset of data to train the model and using the rest for testing. In the testing process we may choose other measures to evaluate a model's performance. These measures are called "model evaluation measures". A common consensus in the machine learning community is that the model selection goal measure and the model evaluation measure should be the same.

In practice we often encounter situations where resources are severely limited, or fast training and testing are required. We only have very limited data for model training and for future performance evaluation, which is called the highly uncertain situations. Naturally one may ask whether the common consensus that the model selection goal measure and the model evaluation measure should be the same is also true under the highly uncertain conditions. Rosset [53] performed initial research on this question with two special measures: accuracy and AUC. He compared the performance of model evaluation measures AUC and accuracy when the model selection goal is accuracy. He showed that AUC can more reliably identify the better model compared with accuracy for Naive Bayes and k-Nearest Neighbor models, even when the model selection goal is accuracy. However, his work has several limitations. First, he only chose very limited data (one synthetic dataset and one real world dataset) to perform the experiment. Second, he did not study model selection with different goals (other than accuracy) using different evaluation measures (other than AUC and accuracy), as learning algorithms and class distributions vary.

In this chapter we thoroughly investigate the problem of model selection under highly uncertain conditions. We analyze the performance of nine different model evaluation

measures under three different model selection goals, four different learning algorithms, on a variety of real world datasets with a wide range of class distributions.

We have obtained some surprising and interesting results. First, we show that the common consensus mentioned above is generally not true under highly uncertain conditions. With the model selection goals of accuracy, AUC or lift, many measures may perform better than these measures themselves. Second, we show that a measure's model selection ability is relatively stable to different model selection goals and class distributions. Third, different learning algorithms call for different measures for model selection.

5.2 Evaluating Model Selection Abilities (MSA) of Measures

We perform experiments to evaluate the model selection abilities of eight commonly used evaluation measures, accuracy (acc), AUC, F-score (FSC), Average Precision (APR), Break Even Point (BEP), Lift, Root Mean Square Error (RMS), Mean Cross Entropy (MXE), and the our new proposed measure SAUC. Details of these measures can be found in Section 1.1 of Chapter 1.

5.2.1 Experiment Process

We perform experiments to simulate model selection tasks under highly uncertain conditions. The goal of these experiments is to study the model selection abilities of measures under different model selection goals, learning algorithms, and class distributions.

In our experiments we choose three model selection goals: accuracy, AUC and lift. Accuracy is chosen because it is the most commonly used measure in a variety of machine learning tasks. Most of the previous research adopted accuracy as the model selection goal [57, 62]. Ranking is increasingly becoming an important task in machine learning. We choose AUC as a model selection goal because it reflects the overall ranking performance of a classifier. Actually AUC has been widely used to evaluate, train and optimize learning algorithms in terms of ranking. We also choose

Table 5.1: Properties of datasets used in experiments

Dataset	Size	Training Size	Attribute #	Class #	Positive Class Ratio
Letter	20000	2000	16	26	50%, 38.2%, 25%, 11.5%, 7.8%, 4%
Adult	30162	4000	14	2	24.8%
Artificial Char	31000	2500	6	10	50%, 30%, 20%, 10%
Chess	28060	2500	6	16	47%, 23.5%, 10%, 5%
Page blocks	5473	1000	10	5	10.2%
Pen digits	10992	1000	16	10	50%, 40%, 30%, 1.4%, 7%, 3%
Nursery	10992	1000	8	5	33.3%
Covtype	29000	2900	54	7	48.8%
Connect-4	38770	3877	42	3	65.8%
Nettalk	20000	1000	3	2	28.2%
Musk	7075	700	50(166)	2	45%
Mushroom	8124	810	22	2	48.2%
Isolet	7797	780	60(617)	26	50%, 38.2%, 25%, 11.5%,7%, 4%
Satimage	6435	640	5	7	9.7%, 23.8%, 30.8%, 47.2%
Phoneme	5427	540	5	2	29.4%
Texture	22000	2200	40	14	36.7%
Ringnorm	7400	740	21	2	27%

lift as another model selection goal because it is very useful in some data mining applications, such as market analysis.

We select 17 large data sets, each with at least 5000 instances. 13 of them are from the UCI repository [4] and the rest are from [15] and [20]. The properties of these datasets are listed in Table 5.1. All multiclass datasets are converted to binary datasets by categorizing some classes to the positive class and the rest to the negative class. For six multiclass datasets, letter, chess, artificial character, pen digits, isolet and satimage, we also vary the class distributions to generate more than one binary datasets. For example, the letter dataset contains 26 classes. We generate 6 different binary datasets with 50%, 38.2%, 25%, 11.5%, 7.8% and 4% of the positive class by selecting the letters of A-M, A-J, A-G, A-C, A-B, A as positive class, respectively. We generate different class distributions because we will investigate whether class distributions influence a measure's model selection ability. From the multiclass datasets we can obtain a total of 41 binary datasets for our experiment as shown in Table 5.1.

We choose four learning algorithms: Support Vector Machine (SVM), k-Nearest Neighbor (KNN), decision trees (C4.5) and Naive Bayes in our study. We choose four different learning algorithms because we want to investigate whether different learning algorithms affect a measure's model selection ability. For each learning algorithm we vary certain parameter settings to generate 10 different learning models with potentially different future predictive performance. For SVM, we choose the

polynomial kernel with the degree of 2 and we vary the regularization parameter C with the values of $10^{-6}, 10^{-5}, \cdots, 1, 10, 50$, and 100. For KNN we set k with different values of 5, 10, 20, 30, 50, 100, 150, 200, 250, and 300. For C4.5 we vary the tree construction stopping parameter $m = 2, 5, 10$ and the tree pruning confidence level parameter $c = 0.1, 0.25, 0.35$. For Naive Bayes we vary the number of attributes of each dataset used to train different learning models. We train a sequence of Naive Bayes models with an increasing number of attributes used, with the attributes of any former model being a subset of any latter model. For example, for the pen digits dataset, we choose the first 1, 2, 4, 6, 8, 10, 12, 14, 15, 16 attributes in training 10 different Naive Bayes models. We use WEKA [64] implementations for these algorithms.

We use the holdout testing method to perform model selection. Our approach is different from the standard cross validation or bootstrap methods. Here only a small sample of the original dataset is used to train learning models, and lots of small test sets are used to simulate the small future unseen data. This is a simple approach to simulate model selection in highly uncertain conditions [53]. Given a model selection goal f, a model evaluation measure g, a learning algorithm and a binary dataset, we use the following experimental process to test the model selection ability of g.

The binary dataset is stratified [1] into 10 equal subsets. One subset is used to train different learning models and the rest are stratified into 100 small equal-sized test sets. We train 10 different learning models of the learning algorithm on the same training subset. For each model we evaluate it on the 100 small test sets. For two models X and Y, X is better than Y iff $E(f(X)) > E(f(Y))$, where $E(f(X))$ is the mean f score measured on X's 100 testing results. g is used to measure X and Y's testing results on each of the 100 testing sets and compare them to see whether or not they agree with $E(f(X))$ and $E(f(Y))$. If f agrees with g then g selects the correct model; otherwise g selects the wrong model. We count in how many cases (among 100) that g selects the correct model. This leads a percentage (or probability) that g can choose the better model between X and Y, representing how well a measure can do in selecting a model. When all pairs of learning models are considered, we use the measure MSA to reflect the overall model selection ability of g. It is defined as

[1] "stratify" means to partition a dataset into some equal-sized subsets with the same class distribution.

$$MSA(g) = \frac{2}{N(N-1)} \sum_{i<j} p_{ij}$$

where N is the number of learning models ($N = 10$), p_{ij} is the probability that measure g can correctly identify the better one from models i and j.

We repeat the above process 10 times by choosing a different subset for training each time. We use the average $MSA(g)$ to measure the model selection ability of g.

We use the MSA measure as the criterion to explore two issues from the experimental results. First, we will compare the MSA of the goal measure with other measures. This will tell us whether it is true that we should always use the model selection goal as the evaluation measure to do model selection. Second, we will explore whether different model selection goals, class distributions and learning algorithms influence a measure's model selection ability.

To clearly explore the above two issues, we need to directly present and analyze the MSA of all the measures in all cases. If a model selection task with a specific model selection goal, dataset, and learning algorithm is called a "model selection case", there are a large number of such model selection cases. One direct approach to clearly show the MSA of different measures is to use a figure to depict the MSA performance for each model selection case.

However, the major problem of this approach is that there are too many such figures to be presented. Since in our experiments we use 41 binary datasets, 4 learning algorithms and 3 model selection goals, there are totally $41 \times 4 \times 3 = 492$ figures. If these figures are categorized according to different model selection goals, there are 164 figures for each model selection goal category. On the other hand, it is also difficult to choose the representative and diverse figures for different model selection cases.

To overcome this difficulty, we use a statistical method to evaluate a measure's MSA. To compare a measure's MSA with that of a model selection goal, we categorize the model selection cases according to different model selection goals. For each model selection case, there is a measure that achieves the best MSA. We compute the percentage of the cases in which one measure can reach the maximum MSA within a varying x% tolerance range, to the total cases. This percentage indicates the success rate that one measure can reach the maximum within an x% range. The success rates

of different measures can be depicted in a figure, in which each curve line represents the success rate of a measure.

5.2.2 Comparing a Measure's MSA with Goal Measure

Figure 5.1(a) depicts the success rates of different measures when we choose accuracy as the model selection goal, while varying the tolerance ranges from 1% to 5%. We can see that the measures SAUC, RMS, MXE, AUC, APR statistically perform better than accuracy for different learning algorithms and datasets. The measures lift and BEP, however, are constantly worse than accuracy.

In Figure 5.1(b) AUC is used as the model selection goal. Only SAUC, RMS and MXE perform better than AUC in most of the sub figures. All other measures are inferior to AUC.

In Figure 5.1(c) lift is used as model selection goal. We can see that except for BEP all measures are constantly better than lift. Furthermore, by comparing Figure 5.1(c) with Figure 5.1(b) and Figure 5.1(a), we can see that the differences of success rates between SAUC, RMS, MXE, AUC, APR with lift are much more than their corresponding differences with accuracy and AUC in Figures 5.1(a) and 5.1(b).

The above discussion shows that under the highly uncertain condition, in general, we should not use the model selection goal measure to perform model selection. This result extends the preliminary work of [53] to more general situations.

5.2.3 The Stability of a Measure's MSA

We next discuss whether one measure's MSA is stable under different model selection goals, class distributions, and learning algorithms.

(i)Model Selection Goals

From the analysis of the previous subsection, we can see that a measure's absolute ability (MSA) is stable to the model selection goals.

(ii)Class Distributions

To explore whether class distributions influence a measure's MSA, we analyze the experimental results according to the datasets with different class distributions. The

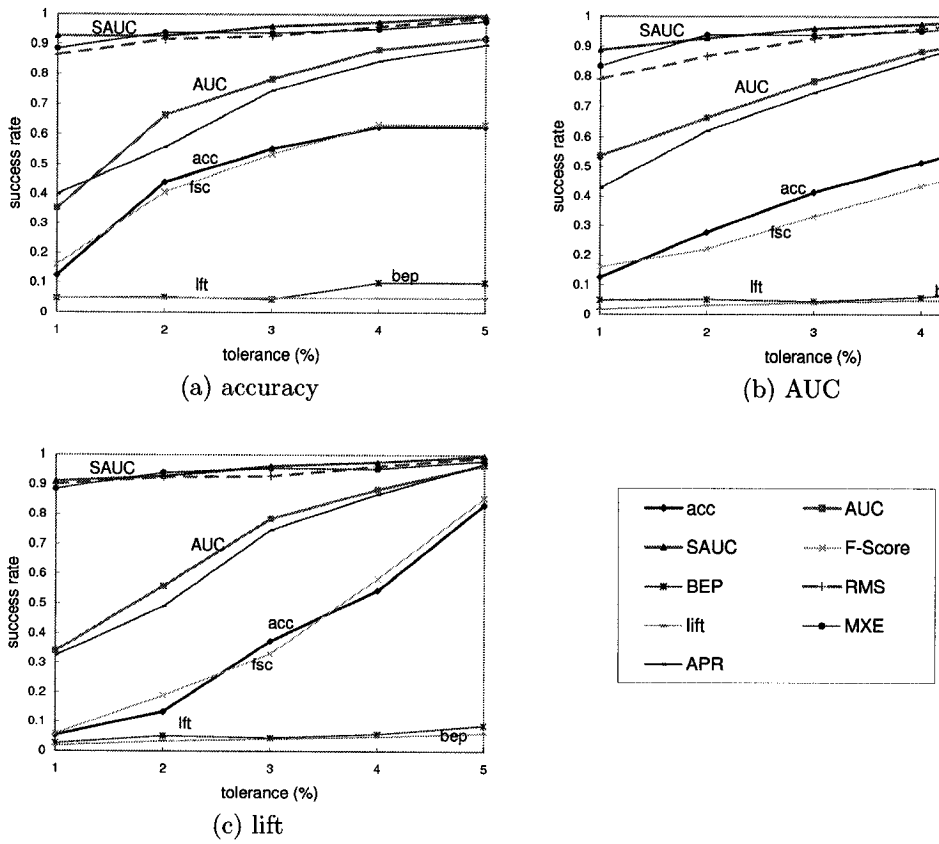
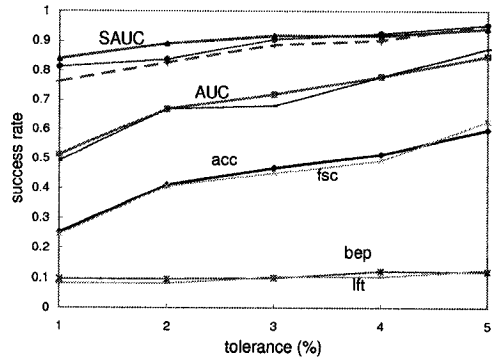


Figure 5.1: Ratio of datasets on which each measure's MSA is within x% tolerance of maximum MSA, using accuracy, AUC and lift as model selection goals.

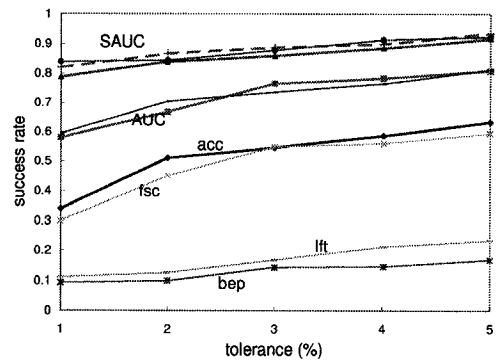
experimental results are categorized into three groups according to the datasets with class distributions of 40%-50%, 25%-30%, 1.4%-10%, respectively. Each group includes the experimental results with all model selection goals and learning models. The success rates of measures are depicted in Figure 5.2. If we rank measures according to their MSA, we can see that generally this ranking is stable to class distributions.

(iii)Learning Algorithms

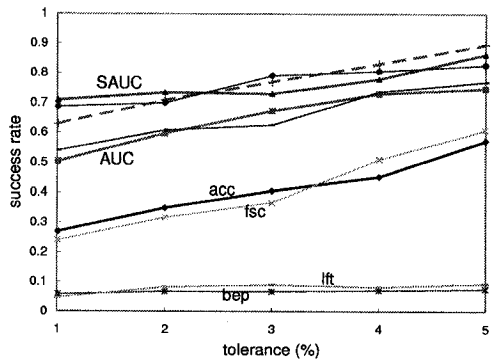
We explore how a measure's MSA is influenced by different learning algorithms. We first discuss how different measures perform for the learning models of SVM and KNN. Here we fix the learning algorithms and vary the datasets and model selection goals. The success rates of measures are depicted in Figure 5.3(a) and Figure 5.3(b) for SVM and KNN, respectively.



(a)datasets with class distributions 40%-50%



(b) datasets with class distributions 25%-30%



(c) datasets with class distributions 1.4%-10%

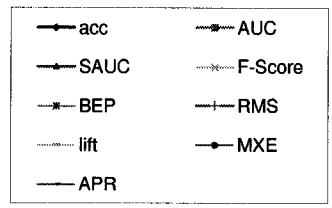
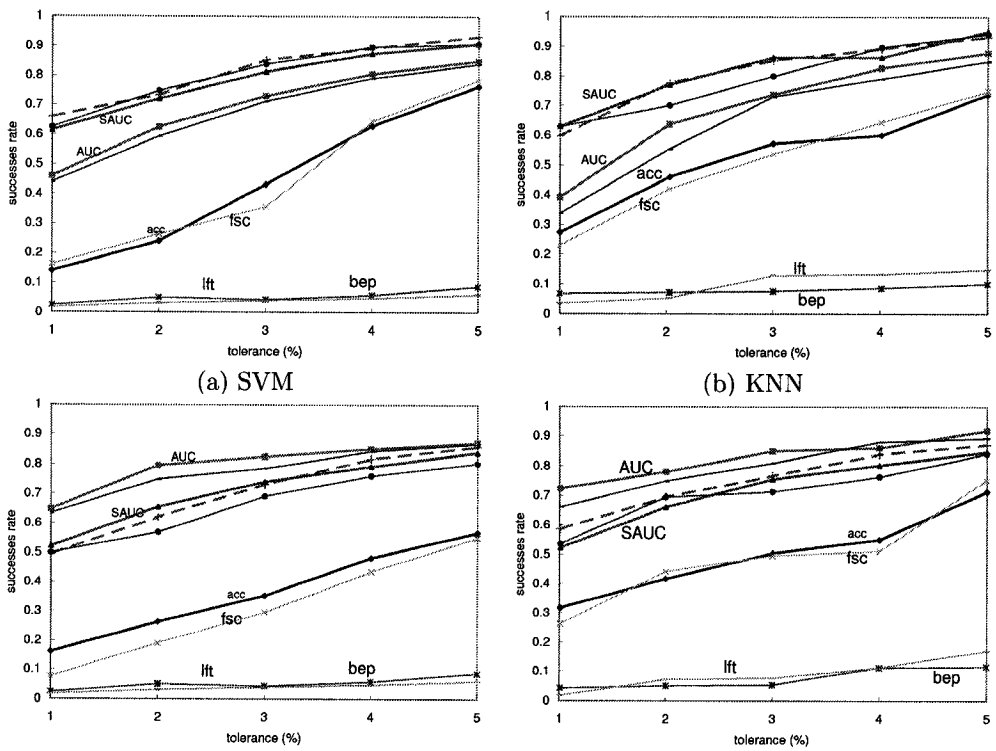


Figure 5.2: Ratio of datasets on which each measure's MSA is within x% tolerance of maximum MSA, for datasets with varied class distributions.

As shown in Figure 5.3(a) and 5.3(b), the measures can be categorized into three different groups according to their performance.

The probability-based measures, including SAUC, RMS and MXE, achieve the best performance. MXE and RMS perform very similarly in most situations. The second group of measures, including AUC and APR, are inferior to the first group measures (SAUC, RMS and MXE). The third group includes the measures of accuracy, F-score, BEP, and lift. This group measures are inferior to the second group measures. F-score is generally competitive with accuracy. Lift and BEP are the two measures always with the worst performance.

Surprisingly, the above three groups of measures match the categories of probability-based measures, ranking measures and threshold measures. Therefore it seems that



(a) SVM (b) KNN

(c) Decision tree (d) Naive Bayes

Figure 5.3: Ratio of datasets on which each measure's MSA is within x% tolerance of maximum MSA, with SVM, KNN, Decision tree and Naive Bayes algorithms.

there is a strong correlation between a measure's category with its model selection ability. An appropriate explanation lies in two aspects. First, the outstanding performance of probability-based measures (RMS, MXE) is partly due to the high quality probability predictions of SVM and KNN learning algorithms. Second, the discriminatory power of the measures also plays an important role. The discriminatory power of a measure reflects how well this measure can discriminate different objects when it is used to evaluate them. Generally a measure's discriminatory power is proportional to the different possible values it can reach. As an example, for a ranked list with n_0 positive instances and n_1 negative instances, accuracy and lift can only reach $n_1 + n_0$ and $(n_0 + n_1)/4$ different values (if we use a fixed 25% percentage for lift). The ranking measure AUC can reach $n_0 n_1$ different values. The probability-based measure RMS, however, can have infinitely many different values. Thus these measures can be ranked according to their discriminatory power (from high to low) as RMS, AUC,

accuracy, lift. This discriminatory power ranking matches with the model selection performance sequence. Therefore we can claim that a measure's model selection ability is closely correlated with its discriminatory power for the SVM and KNN learning algorithms. The possible reason is that a measure with high discriminatory power usually uses more information in evaluating objects and thus is more robust and reliable. Probability-based measures use the predicted probability information, and thus they are more accurate than ranking measures which only use the relative ranking position information. Similarly, ranking measures also use more information than accuracy or lift, which only considers the classification correctness in the part or whole dataset ranges.

However, compared with SVM and KNN learning algorithms, measures perform differently for decision trees (C4.5) and Naive Bayes. The success rate graphs are shown in Figure 5.3(c) and Figure 5.3(d) for Naive Bayes and decision trees. We can see that probability-based measures do not always perform better than ranking measures. This indicates that they might be unstable for some datasets and model selection goals. By comparing ranking measures with threshold measures, however, we can see that these two kinds of measures are less influenced by learning algorithms. We can conclude that generally the measures of RMS, SAUC, MXE, AUC, APR have the best performance for decision trees (C4.5) and Naive Bayes algorithms.

[17, 50] have shown that learning algorithms of C4.5 and Naive Bayes usually produce poor probability estimations. The poor probability estimations directly degrade the performance of SAUC, RMS and MXE when they are used to rank learning models. This explains why the probability-based measures perform unstably for C4.5 and Naive Bayes models. Although the poor probability estimations also influence the ranking measures of AUC and APR, these influences are not so strong. This also explains why the ranking measures perform relatively stably.

In summary, from the above discussions we can draw the following conclusions.

1. For model selection tasks under the highly uncertain conditions, the common consensus that the goal measure should be used to do model selection is not true.

2. A measure's model selection performance is relatively stable to the selection goals and class distributions.

3. Different learning algorithms need to choose different measures for model selection tasks. For learning algorithms with good quality of probability predictions (such as SVM and KNN) a measure's model selection ability is closely correlated with its discriminatory power. The probability-based measures (SVM, SAUC, MXE) perform best, followed by ranking measures (AUC, APR), followed by threshold measures (Accuracy, FSC, BEP, lift). For learning algorithms with poor probability predictions (such as C4.5 and Naive Bayes), the probability-based measures such as SVM, SAUC and MXE perform quite unstably. AUC and Average Precision become robust and well performed measures.

5.3 Summary

Model selection is a significant task in machine learning and data mining. In this chapter we perform a thorough empirical study to investigate how different measures perform in model selection under highly uncertain conditions, with varying learning algorithm, model selection goals and dataset class distributions. We show that a measure's model selection performance is relatively stable by model selection goals and class distributions. However, different learning algorithms call for different measures for model selection.

For our future work, we plan to investigate model selection tasks under other uncertain conditions. We also plan to devise new model selection measures that are specialized under different conditions.

Chapter 6

Conclusions and Future Work

In this chapter, we will summarize our work and main contributions, and give several related research directions for the future work.

6.1 Contributions

In this thesis, we addressed three major issues: Comparing machine learning measures (Chapter 3), constructing better machine learning measures (Chapter 4) and applying measures to model selection (Chapter 5). For the first issue, we theoretically and empirically compared the measures of AUC and accuracy, and some ranking measures. More specifically, our major contributions in this aspect are listed below.

- We formally proposed five percentage criteria and two degree criteria with the goal to provide a detailed and complete comparison between two arbitrary one-number measures. This is significant since it provided a general approach to evaluate the relative measure performance.
- We used the criteria proposed to give a complete and detailed comparison between the measures of AUC and accuracy. We formally proved that AUC is indeed consistent to, and more discriminant measure than accuracy. We also performed experiments by using artificial and real-world datasets to confirm the theoretical results. Finally, we reevaluated some popular machine learning algorithms with AUC and obtained some new and interesting results.

- We used the proposed criteria to compare six ranking measures. We also proposed a new ranking measure called OAUC. From the experiment with artificial datasets we computed the degree of consistency and discriminancy between every two ranking measures. This led to a preference order of the ranking measures assessed. An additional experiment with real-world datasets and ranking algorithms confirmed our conclusion about the preference order of ranking measures.

For the second aspect, concerned with the construction of better measures, we focused on building new measures from existing ones. More specifically, our major contributions are:

- We proposed two general approaches to construct new measures from two existing measures. The two methods are linear combination and two-level construction. This work is important since the construction approaches are general methods that are independent of specific domains or applications.

- We theoretically analyzed the consistency and discriminancy relationships between the new constructed measures and the original measures. We both formally and empirically showed that the new measures are consistent to, and more discriminant than the original ones.

- By using AUC and accuracy we constructed two kinds of new measures. We compared these new measure to another robust and well performed measure: RMS. We showed that the two new measures are both more closely correlated with RMS than that of AUC, and AUC is more correlated with RMS than accuracy. We concluded that the new measures are finer than AUC and accuracy.

- We used the new two-level measure, AUC and accuracy to train the Artificial Neural Networks, respectively, to obtain different learning models. We also used the three different measures to evaluate the learning models. The experimental results showed that ANN trained by the new two-level measure performs better than the learning model trained by AUC, and they are both better than the model trained with accuracy when evaluated with all three different measures.

For the third issue, we investigated the application of using measures in model selection tasks. Our major contributions are:

- We evaluated the model selection abilities of nine performance measures under the highly uncertain condition with different model selection goals, learning algorithms, and class distributions. We showed that to achieve better model selection performance, generally we should not use the goal measure to select model.
- We showed that the model selection ability of a measure is stable to different model selection goals and dataset class distributions. However, different learning algorithms call for different measures for model selection.
- We showed that for learning algorithms of SVM and KNN, generally the measures of RMS, SAUC,MXE are preferred for model selection. For learning algorithms of decision trees and naive Bayes, generally the measures of RMS, SAUC, MXE, AUC, APR are preferred.

6.2 Future Work

We have investigated the issues of measures comparison, new measures construction, and model selection with measures. The goal of this research is to explore the possibilities of using different or new constructed measures to improve the performance of learning algorithms. Some work has been done in this direction.

There are several approaches for our future work.

First, we plan to modify the traditional classification algorithms for the purpose of ranking. By modifying the construction criteria, structures, parameters and so on, some traditional classification algorithms, such as Bayesian Networks, can produce more satisfactory ranking performance. We have done some work on improving the ranking performance for some popular learning algorithms. For example, we proposed a novel dynamic ensemble re-construction method that aims at improving the ranking performance of a given ensemble.

Second, we want to reconstruct or optimize different learning algorithms with new heuristic measures. Traditionally, different machine learning algorithms are constructed with different heuristic measures. For example, decision trees are built with the entropy-based measures; neural networks are trained with least squared errors. One natural question is how these learning algorithms would perform if they are

constructed with different measures? Some research has been done in this direction. The measure of AUC is used to build decision trees and Bayesian Networks [22, 52]. These new learning algorithms show better classification performance compared with the originals. We will use the proposed new machine learning measures to train and optimize different learning algorithms.

References

[1] George E. Andrews. *The Theory of Partitions.* Addison-Wesley Publishing Company, 1976.

[2] George E. Andrews. On the difference of successive gaussian polynomials. *Journal of Statistical Planning and Inference*, 34:19–22, 1993.

[3] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-D point sets. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[4] C.L. Blake and C.J. 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.

[5] C.L. Blake and C.J. Merz. *UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html].* Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[6] 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.

[7] 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.

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

[9] Rich Caruana and Alexandru Niculescu-Mizil. Data mining in metric space: An empirical analysis of supervised learning performance criteria. In *Proceedings of the 10th ACM SIGKDD conference*, 2004.

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

[11] M. P. Clements and D. F. Hendary. On the limitations of comparing mean square forcast errors. *Journal of Forcasting*, 12(9):617–637, 1993.

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

[13] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

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

[15] Delve. Delve project: Data for evaluating learning in valid experiments. http://www.cs.toronto.edu/ delve/, 2003.

[16] 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.

[17] P. Domingos and M. Pazzani. Beyond independence: Conditions for the optimality of the simple Bayesian classifier. *Machine Learning*, 29:103–130, 1997.

[18] Richard O. Duda and Peter E. Hart. *Pattern classification and scene analysis.* A Wiley-Interscience Publication, 1973.

[19] J.P. Egan. *Signal Detection Theory and ROC analysis.* Academic Press, New York, 1975.

[20] Elena. Elena datasets. ftp://ftp.dice.ucl.ac.be/pub/neural-nets/ELENA/databases, 1998.

[21] 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.

[22] 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.

[23] P.A. Flach. The geometry of ROC space: understanding machine learning metrics through ROC isometrics. In *Proc. 20th International Conference on Machine Learning (ICML'03)*, 2003.

[24] D.M. Green and J.A. Swets. *Signal Detection Theory and Psychophysics.* Wiley, New York, 1966.

[25] 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.

[26] J. A. Hanley and B. J. McNeil. The meaning and use of the area under a receiver operating characteristics (ROC) curve. *Radiology*, 143:29–36, 1982.

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

[28] C.W. Hsu and C.J. 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.

[29] J. Huang and C. X. Ling. Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310, 2005.

[30] J. Huang and C.X. Ling. Partial ensemble classifiers selection for better ranking. In *Proceedings of the 5th IEEE International Conference on Data Mining*, 2005.

[31] J. Huang and C.X. Ling. Evaluating model selection abilities of performance measures. In *Proceedings of the Workshop on Evaluation Methods for Machine Learning at the 21st National Conference on Artificial Intelligence (AAAI-06)*, 2006.

[32] 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.

[33] F. Jensen. *An introduction to Bayesian Networks.* UCL Press, 1996.

[34] J. F. Kenney and E. S Keeping. *Mathematics of Statistics.* Princeton, NJ, 1962.

[35] 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.

[36] 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.

[37] 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 519–526, 2003.

[38] C. X. Ling, J. Huang, and H. Zhang. AUC:A better measure than accuracy in comparing learning algorithms. In *Proceedings of 16th Canadian Conference on Artificial Intelligence*. Springer, 2003. To appear.

[39] 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.

[40] C.X. 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.

[41] H. Linhart and W. Zucchini. *Model Selection*. New York:Wiley.

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

[43] C.E. Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medcine*, 8:283–298, 1978.

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

[45] M. J. Pazzani. Search for dependencies in Bayesian classifiers. In D. Fisher and H. J. Lenz, editors, *Learning from Data: Artificial Intelligence and Statistics V*. Springer Verlag, 1996.

[46] Judea Pearl. *Probabilistic reasoning in intelligent systems:networks of plausible inference*. Morgan Kauhmann, 1988.

[47] F. Provost and P. Domingos. Well-trained PETs: improving probability estimation trees. In *Technical Report CDER #0004-IS, Stern School of Business, New York University*. http://www.stern.nyu.edu/ fprovost/, 2000.

[48] F. Provost and P. Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52:3:199–215, 2003.

[49] 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.

[50] 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.

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

[52] Alain Rakotomamonjy. Optimizing AUC with SVMs. In *Proceedings of European Conference on Artificial Intelligence Workshop on ROC Curve and AI*, 2004.

[53] Saharon Rosset. Model selection via the AUC. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

[54] Timothy W. Ruefli and Robert R. Wiggins. When mean square error becomes variance: A comment on "business risk and return: A test of simultaneous relationships". *Management Science*, 40(6):750–759, 1994.

[55] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986.

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

[57] D. Schuurmans. A new metric-based approach to model selection. In *Proceedings of National Conference on Artificial Intelligence(AAAI-97)*, 1997.

[58] 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.

[59] K.A. Spackman. Signal detection theory: Valuable tools for evaluating inductive learning. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 160–163, 1989.

[60] 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.

[61] J.A. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240:1285–1293, 1988.

[62] Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data.* Springer-Verlag NY, 1982.

[63] Vladimir Vapnik. *Statistical Learning Theory.* Wiley, NY, 1998.

[64] Ian H. Witten and Eibe Frank. *Data Mining: Practical machine learning tools with Java implementations.* Morgan Kaufmann, San Francisco, 2000.

[65] Lian Yan, Robert Dodier, Michael C. Mozer, and Richard Wolniewicz. Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *Proceedings of the 20th International Conference on Machine Learning*, pages 848–855, 2003.

Vita

Name	Jin Huang
Post-secondary Education and Degrees	Tianjin University Tianjin, China 1988–1992 B.Sc.
	Shanghai Jiaotong University Shanghai, China 1992–1995 M.Sc.
	The University of Western Ontario London, Canada 2002–2006 Ph.D.
Honours and Awards	Special University Scholarship, 2002-2006 Teaching Assistantship Award, 2002-2006 Research Assistantship Award, 2002-2006 Graduate Tuition Scholarship, 2002-2006
Related work experience	Research Assistant University of Western Ontario, Ontario, Canada 2002–2006
	Teaching Assistant University of Western Ontario, Ontario, Canada 2002–2006

Publications

1. Jin Huang, Charles Ling, "Evaluating Model Selection Abilities of Performance Measures", In the Workshop on Evaluation Methods for Machine Learning at the 21st National Conference on Artificial Intelligence (AAAI-06), Boston, 2006.

2. Jin Huang, Charles Ling, "Using AUC and Accuracy in Evaluating Learning Algorithms". IEEE Transactions on Knowledge and Data Engineering (TKDE), vol. 17, no. 3, pp. 299-310, March, 2005.

3. Jin Huang, Charles Ling, "Partial Ensemble Classifiers Selection for Better Ranking". Proceedings of the 5th IEEE International Conference on Data Mining (ICDM2005), Houston, USA, Nov. 2005.

4. Jin Huang, Charles Ling, "Dynamic Ensemble Re-Construction for Better Ranking". Proceedings of 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), Porto, Portugal, Oct. 2005.

5. Jin Huang, Charles Ling, "Rank Measures for Ordering". Proceedings of 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD), Porto, Portugal, Oct. 2005.

6. Charles Ling, Jin Huang, and Harry Zhang, "AUC: a Statistically Consistent and more Discriminating Measure than Accuracy". Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI), Acapulco, Mexico, Aug. 2003.

7. Jin Huang, J. Lu, and Charles Ling, "Comparing Naive Bayes, Decision Trees, and SVM using Accuracy and AUC". Proceedings of Third IEEE International Conference on Data Mining (ICDM'2003), Melbourne, Florida, USA, Dec. 2003.

8. Charles Ling, Jin Huang, and Harry Zhang. "AUC: a Better Measure than Accuracy in Comparing Learning Algorithms". Proceedings of 2003 Canadian Artificial Intelligence Conference, Halifax, Canada, Jun. 2003