ReverseTesting: An Efficient Framework to Select Amongst Classifiers under Sample Selection Bias

Wei Fan
IBM T.J.Watson Research
Hawthorne, NY 10532, USA
weifan@us.ibm.com

Ian Davidson
Department of Computer Science, University of Albany, State University of New York
Albany, NY 12222, USA
davidson@cs.albany.edu

ABSTRACT

Perhaps, one of the most important assumptions made by classification algorithms is that the training and test sets are drawn from the same distribution, i.e., the so-called "stationary distribution assumption" that the future and the past are identical from a probabilistic standpoint. In many domains of real-world applications, such as marketing solicitation, fraud detection, drug testing, loan approval among others, this is rarely the case. This is because the only labeled sample available for training is biased due to a variety of practical reasons. In these circumstances, traditional methods to evaluate the expected generalization error of classification algorithms, such as structural risk minimization, ten-fold cross-validation, and leave-one-out validation, usually return poor estimates of which classification algorithm will be the most accurate. Sometimes, the estimated order of the learning algorithms' accuracy is so poor that it is no better than random guessing. Therefore, a method to determine the most accurate learner is needed for data mining under sample selection bias. We present such an approach that can determine which learner will perform the best on an unbiased test set, given a possibly biased training set, in a fraction of the cost to use cross-validation based approaches.

Keywords: Classification, Sample Selection Bias, Stationary Distribution Assumption.

1. INTRODUCTION

Consider the following typical situation a data mining practitioner faces. He or she has been given a training set and is asked to build a highly accurate predictive model that will be applied to make a prediction on some future set of testing instances. The practitioner has at his or her disposal a variety of algorithms to learn classifiers, such as decision trees, naive Bayes and support vector machines, and wishes to determine the best performing algorithm. The standard approach to determine the most accurate algorithm is to perform cross-validation or leave-one out validation on the training set or if the Vapnik-Chervonenkis dimension of the model space is known, to perform structural risk minimization [Shawe-Taylor et' al 1996]. These standard approaches have served the data mining and machine learning community well.

However, as data mining algorithms are applied to more challenging domains, the assumptions made by traditional algorithms are violated. Perhaps one of the strongest assumptions made by classification algorithms is known as the "stationary distribution assumption" in the machine learning literature [Vapnik 1995] and "non-biased distribution" in the data mining literature [Zadrozny, 2004].

DEFINITION 1.1. ***Stationary or Non-Biased Distribution Assumption*** *[Vapnik 1995] Each and every training set instance and test set instance is identically and independently drawn from a* ***common*** *distribution* $\mathcal{Q}(\mathbf{x}, y)$.

However, consider the example, where we are asked to build a model to predict if a particular drug is effective for the entire population of individuals, that is, instances in the future test set will be an unbiased sample. However, the available training data is typically a sample from previous hospital trials where individuals self select to participate and are representative of the patients at that hospital but not of the entire popular [Zadrozny, 2004]. In the application of data mining to direct marketing, it is common practice to build models of the response of customers to a particular offer using only the customers that have received the offer in the past as the training set, and then to apply the model to the entire customer database. Because these offers are usually not given at random, the training set is not drawn from the same population as the test set. Therefore, a model constructed using this training set may not perform well for the entire population of customers.

In this paper we relax the stationary distribution assumption and instead allow the training set and test set to be drawn from differing distributions but within some weak limitations.

DEFINITION 1.2. ***Biased Training Instance Distribution Assumption*** *Each and every training instance is drawn from distribution* $\mathcal{P}(\mathbf{x}, y)$*, and test set instance is identically and independently drawn from distribution* $\mathcal{Q}(\mathbf{x}, y)$*.* $\mathcal{Q}(\mathbf{x}, y)$ *is the true unbiased distribution of instances,* $\mathcal{P}(\mathbf{x}, y)$ *is a biased distribution, and* $\mathcal{Q}(\mathbf{x}, y) \neq \mathcal{P}(\mathbf{x}, y)$.

The above definition only states that $\mathcal{P}(\mathbf{x}, y) \neq \mathcal{Q}(\mathbf{x}, y)$, but the two distributions may still differ in many different ways. The framework presented by Zadrozny [Zadrozny, 2004] discusses various types of bias. For example, one of the prevalent bias is "feature bias", which is best understood via the standard decompositions $\mathcal{P}(\mathbf{x}, y) = \mathcal{P}(\mathbf{x}) \cdot \mathcal{P}(y|\mathbf{x})$ and $\mathcal{Q}(\mathbf{x}, y) = \mathcal{Q}(\mathbf{x}) \cdot \mathcal{Q}(y|\mathbf{x})$. Feature bias can occur when $\mathcal{P}(\mathbf{x}) \neq \mathcal{Q}(\mathbf{x})$ but $\mathcal{P}(y|\mathbf{x}) = \mathcal{Q}(y|\mathbf{x})$ (details in Section 2). An example of feature bias is the drug modeling

KDD'06, August 20-23, 2006, Philadelphia, PA USA

example given earlier. The concept of when the drug is effective does not change between the training and test sets, only the chance of encountering a feature vector (representative of a person in this situation) is different in the chosen hospital from the general population.

When the assumption of stationary distribution is lifted, it raises problems for answering the question: "Which classification algorithm finds the best performing model?" As we shall see the traditional approaches which are used extensively by the data mining community, such as cross-validation and leave-one-out validation, perform hopelessly when sample bias occurs. In some circumstances, the order of expected accuracy of competing models is not even better than random guessing.

Previous work on this problem by Zadrozny [Zadrozny, 2004] noted that some learners, such as logistic regression and hard-margin support vector machines, are invariant to feature bias and describes how to correct this type of sample bias for those learners that are sensitive to feature bias, such as decision trees and naive Bayes. However, this work is limited to situations where one could build a model that is asymptotically close to the true unbiased model $\mathcal{Q}(y|\mathbf{x})$. Recently however, Fan, Davidson, Zadrozny and Yu [Fan' et al 2005] illustrated that this is not always possible, and all types of learner may be effected by feature sample bias. It is difficult to know which algorithm is not affected by bias without knowing the true model $\mathcal{Q}(y|\mathbf{x})$. Importantly however, the true model $\mathcal{Q}(y|\mathbf{x})$ is generally never known for real-world problems. That is, we cannot apply some types of learners and assume that they will be able to overcome sample bias. Given this earlier result, the problem associated with learning with a biased sample is:

PROBLEM 1.1. ***The Learning From Sample Biased Problem*** *Given a labeled training set D, an unlabeled test set T, such that D and T may or may not be drawn from the same distribution, and a series of learning algorithms ($L_1 ... L_k$): Which learner when applied to D generates the model that is most accurate on T?*

We begin by discussing various types of sample selection bias as well as the notations used throughout the paper. In Section 3, we empirically show that traditional approaches, cross-fold validation and leave-one-out validation on the training set, can give misleading, sometimes pessimistic, solutions to Problem 1.1. In particular, we provide an explanation of their poor performance in Section 3.4. In Section 4, we describe and explain the mechanism of the proposed algorithm to solve Problem 1.1. Section 5 empirically illustrates that our algorithm outperforms the traditional approaches. To be exact, with both biased and unbiased datasets, we show that our algorithm is capable of choosing the best learner 84% of the time while cross-validation and leave one-out validation achieve rates approximately from 40% to 68%. Importantly, in Section 6, we have applied the proposed approach on charity donation solicitation and credit card card fraud detection datasets, where sample selection bias is common. The proposed method correctly ordered performance of all competing classifiers, while cross-validation was right 58% of the time. In Section 7, we explain the mechanism of the proposed algorithm in terms of matching true probability distributions. Related work on sample selection bias is reviewed in Section 8.

2. TYPES OF SAMPLE SELECTION BIAS AND NOTATION

Assume that the event $s = 1$ denotes that a labeled training example $(\mathbf{x}, y)$ is selected from the unbiased joint distribution $\mathcal{Q}(\mathbf{x}, y)$ of examples into the training set D, and that $s = 0$ denotes that $(\mathbf{x}, y)$ is not chosen. Using the dependency on s, the training set is sampled from the distribution $\hat{\mathcal{P}}(\mathbf{x}, y, s = 1)$. Since $s = 1$ is a fixed value, we can simplify this notation by removing the explicit dependency on $s = 1$, and it becomes $\mathcal{P}(\mathbf{x}, y)$. In other words, we define $\hat{\mathcal{P}}(\mathbf{x}, y, s = 1) = \mathcal{P}(\mathbf{x}, y)$.

The training distribution $\mathcal{P}(\mathbf{x}, y)$ and testing distribution $\mathcal{Q}(\mathbf{x}, y)$ are related by $\mathcal{P}(\mathbf{x}, y) = \hat{\mathcal{P}}(\mathbf{x}, y, s = 1) = \mathcal{Q}(\mathbf{x}, y) \cdot \hat{\mathcal{P}}(s = 1|\mathbf{x}, y)$. This is straightforward by applying the product rule, such that $\hat{\mathcal{P}}(\mathbf{x}, y, s = 1) = \hat{\mathcal{P}}(\mathbf{x}, y) \cdot \hat{\mathcal{P}}(s = 1|\mathbf{x}, y)$. As sample selection bias is denoted through $s = 1$, $\hat{\mathcal{P}}(\mathbf{x}, y)$ is the same as the true unbiased distribution or $\hat{\mathcal{P}}(\mathbf{x}, y) = \mathcal{Q}(\mathbf{x}, y)$. In addition, $\hat{P}(s = 1|\mathbf{x}, y)$ is equivalent to $P(s = 1|\mathbf{x}, y)$, as introduced by Zadrozny in [Zadrozny, 2004].

In [Zadrozny, 2004], four different types of sample selection bias are clearly discussed according to the dependency of s on $\mathbf{x}$ and y. Note that in all cases the test set examples are assumed to be unbiased, since the model will be used on the entire population. A summary of all notations and assumptions made in this paper is in Figure 2.

In the **complete independent case** s is independent from both $\mathbf{x}$ and y, i.e., $P(s = 1|\mathbf{x}, y) = P(s = 1)$. That is, the sample selection bias depends on some other event completely independent of the feature vector $\mathbf{x}$ and the true class label y.

In the **feature dependent case** or **feature bias case**, the selection bias s is dependent on the feature vector $\mathbf{x}$ and is *conditionally* independent of the true class label y given $\mathbf{x}$, i.e., $P(s = 1|\mathbf{x}, y) = P(s = 1|\mathbf{x})$. The training distribution $\mathcal{P}(\mathbf{x}, y) = \mathcal{P}(\mathbf{x})\mathcal{P}(y|\mathbf{x})$ and test distribution $\mathcal{Q}(\mathbf{x}, y) = \mathcal{Q}(\mathbf{x})\mathcal{Q}(y|\mathbf{x})$ are associated via $\mathcal{P}(\mathbf{x}) = \mathcal{Q}(\mathbf{x}) \cdot P(s = 1|\mathbf{x})$ and $\mathcal{P}(y|\mathbf{x}) = \mathcal{Q}(y|\mathbf{x})$. This type of sample selection is extensive in many mining applications. For example, in the direct marketing case mentioned earlier, the customers are selected into the training sample based on whether or not they have received the offer in the past. Because the decision to send an offer is based on the known characteristics of the customers (that is, $\mathbf{x}$) before seeing the response (that is, y) then the bias will be of this type. This type of bias also occurs in medical data where a treatment is not given at random, but the patients receive the treatment according to their symptoms which are contained in the example description (i.e., the $\mathbf{x}$ values). Therefore, the population that received the treatment in the past is usually not a random sample of the population that could potentially receive the treatment in the future.

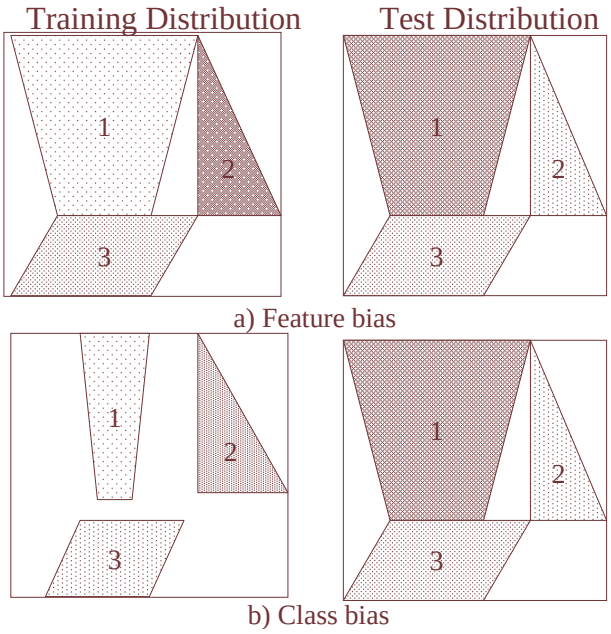
In the **class dependent case**, the selection bias is dependent only on the true class label y, and is *conditionally* independent from the feature vector $\mathbf{x}$, i.e., $P(s|\mathbf{x}, y) = P(s|y)$. This occurs when there is a correlation between the label value and the chance of appearance in the database. For example, people with higher income may be less inclined to answer a survey about income. Thus, if we are trying to learn to predict an individual's income category using survey data, class dependent bias is likely to occur.

In the **complete dependent case**, there is no assumption about any restriction of the independence of s given $\mathbf{x}$ and y, and both the example description and its label influence whether the example will be chosen into the training set.

2.1 Effect of Bias on Learning

Figure 1 illustrates the effect of feature as well as class bias on the training and test set distributions. In situation a), since feature bias cannot change $P(y|\mathbf{x})$ but only $P(\mathbf{x})$, the class boundaries typically do not change unless $P(\mathbf{x})$ becomes zero for some areas of the instance space. We see that in the training set, the true probability $P(y|\mathbf{x})$ is under-estimated in region 1, over-estimated

Figure 1: Visualization of a possible effect of a) feature bias and b) class bias on training set. There are two classes, and only areas with positive class "+" are shaded, and the darkness or intensity shows frequency of examples in the highlighted region.



in region 2, and, for region 3, there is no effect. In situation b), class bias can change the class boundaries. The positive class or P(y="+") is under estimated in the training set, and hence the positive regions shrink in size and "intensity".

For the remainder of this paper, we concentrate on feature bias as it is believed to happen extensively in practice [Zadrozny, 2004, Fan' et al 2005]. In addition, we show below that when the prior class probability is the same in the training and test sets or $P(y|s=1)=P(y)$, the only possible sample selection bias is feature bias or $P(s=1|\mathbf{x},y)=P(s=1|\mathbf{x})$.

$$
\begin{aligned}
& P(s=1|\mathbf{x},y) && \\
&= \frac{P(y,\mathbf{x}|s=1)P(s=1)}{P(\mathbf{x},y)} && \text{Bayes Theorem} \\
&= \frac{P(\mathbf{x}|s=1).P(y|\mathbf{x},s=1)P(s=1)}{P(\mathbf{x},y)} && \text{Product Rule} \\
&= \frac{P(\mathbf{x}|s=1).P(y|\mathbf{x})P(s=1)}{P(\mathbf{x},y)} && \text{No Class Bias} \\
&= \frac{P(\mathbf{x},s=1).P(\mathbf{x},y)P(s=1)}{P(s=1)P(\mathbf{x})P(\mathbf{x},y)} && \text{Conditional Probability} \\
&= \frac{P(\mathbf{x},s=1)}{P(\mathbf{x})} && \text{Cancellation} \\
&= P(s=1|\mathbf{x}) && \text{Conditional Probability}
\end{aligned}
$$

3. FAILURE OF TRADITIONAL APPROACHES

We begin this section by introducing the eleven data sets used throughout the paper.

3.1 Biased and Unbiased Datasets

Since in our learning problem definition (see Problem 1.1) we do not explicitly know if the training sample is biased, we include data sets with little or no sample bias such as several Newsgroup data sets in addition to the UCI data sets that will be purposefully

- $\mathbf{x}$ is feature vector, y is class label, and $s=1$ denotes that an example $(\mathbf{x},y)$ is selected into the training set D.
- Sample selection bias is formalized as dependency between $s=1,\mathbf{x}$ and y, as $P(s=1|\mathbf{x},y)$. Different types of sample selection bias can be found in Section 2.
- In particular, feature bias is denoted as $P(s=1|\mathbf{x},y)=P(s=1|\mathbf{x})$.
- $\mathcal{Q}(\mathbf{x})$ is the true target probability of feature vector $\mathbf{x}$ in the instance space.
- $\mathcal{Q}(y|\mathbf{x})$ is the true conditional probability for feature vector $\mathbf{x}$ to have class label y.
- The test set T is drawn from the unbiased joint distribution $\mathcal{Q}(\mathbf{x},y)=\mathcal{Q}(\mathbf{x})\mathcal{Q}(y|\mathbf{x})$. But since the labels are not available, we only have $T=(X)=\{\mathbf{x}_i\}$.
- The training set $D=(X,Y)=\{(\mathbf{x}_i,y_i)\}$ is drawn from the joint distribution $\hat{\mathcal{P}}(\mathbf{x},y,s=1)$.
- When the explicit dependency on $s=1$ is omitted in the notation, the distribution $\hat{\mathcal{P}}(\mathbf{x},y,s=1)$ is short-handed as $\mathcal{P}(\mathbf{x},y)$, and $\mathcal{P}(\mathbf{x},y)=\mathcal{P}(\mathbf{x})\cdot\mathcal{P}(y|\mathbf{x})$.
- When $s=1$ is elaborated, $\hat{\mathcal{P}}(\mathbf{x},y,s=1)$ is decomposed into $\mathcal{Q}(\mathbf{x},y)\cdot P(s=1|\mathbf{x},y)$, or a product of true unbiased joint distribution with sample selection bias.
- $\mathcal{P}(\mathbf{x})$ and $\mathcal{Q}(\mathbf{x})$ are the probability distributions of feature vectors in the training and test sets respectively.
- $\mathcal{P}(\mathbf{y}|\mathbf{x})$, $\mathcal{Q}(\mathbf{y}|\mathbf{x})$ is the conditional probability distribution of class labels given the feature vectors.
- Under feature bias, $\mathcal{P}(y|\mathbf{x})=\mathcal{Q}(y|\mathbf{x})$ and $\mathcal{P}(\mathbf{x})=\mathcal{Q}(\mathbf{x})P(s=1|\mathbf{x})$.
- Θ is the model space assumed by a learner.
- θ_a is a best model found by learning algorithm L_a by searching in its chosen model space Θ_a given training data D.
- $T_a=\{(\mathbf{x},y_a)\}$ is the labeling of the test set T given by the classifier θ_a.
- $\mathcal{Q}_a$ is θ_a's estimate of $\mathcal{Q}(\mathbf{x},y)$.
- θ_b^a is a new classifier built from T_a using learner L_b, that is the model built by L_b using the test set labeled by L_a.
- $AccTrain_b^a$ is the accuracy on the training set D of the classifier θ_b^a built from T_a using learner L_b. It is not the typical training set accuracy.

Figure 2: Summary of Symbols and Concepts

biased.

We perform experiments on articles drawn from six pairs of Newsgroups [Rennie 2003]. In half of these problems (Mac-Hardware, Baseball-Hoc, and Auto-space), the articles in the newsgroups are very similar, and in the other half (Christ-Sales, MidEast-Elec, and MidEast-Guns), are quite different. The training and test datasets are created using the standard bydate division into training (60%) and test (40%) based on the posting date. This division potentially creates a sample bias. For example, in the MidEast-Guns newsgroup, the word "Waco" occurs extensively in articles in the training set but not in the test set, as interest in the topic fades. Therefore, instances of articles containing the word "Waco" in the training set are much more populous than in the test set. Since the proportion of each class label is the same in the training and test data sets, there is no class label bias. We used Rainbow [McCallum 1998] to extract features from these news articles. The feature vector for a document consists of the frequencies of the top ten words by selecting words with highest average mutual information with the class variable.

The UCI data sets are feature biased by sorting the training set on the first attribute and removing the first 25% of records thereby creating a selection bias based on the first feature. The test sets are unaltered.

3.2 Traditional Approaches

For each data set we attempt to estimate the generalization error from the training set for a variety of learners to determine the best performing learner. There are two common areas/approaches to achieve this. The structural risk minimization approach bounds the generalization error as a function of the training set error and Vapnik Chervonenkis (VC) dimension. Formally, $GE \leq TE + \sqrt{\frac{VC(L)log(\frac{2n}{VC(L)}+1)-log(\frac{\delta}{4})}{n}}$, where GE is the generalization error, TE is the training set error, $VC(L)$ is the VC dimension of the learner L, n is the size of the training set and δ is the chance of the bound failing. However, this bound derivation explicitly makes the stationary distribution assumption and makes no claim to formally hold when it is violated as in our case [Vapnik 1995].

Two empirical alternatives for generalization error estimation commonly used in data mining is ten-fold cross-validation and leave-one-out validation. In the ten-fold cross-validation approach, the training data set is divided into ten equally sized, randomly chosen folds. Each fold is used to evaluate the accuracy of a model built from the remaining nine folds, the average accuracy on the hold-out fold is then used as an estimate of generalization error. Typically, as in our experiments, the entire process is repeated one hundred times with different randomly generated folds. With the leave-one-out validation approach, each instance is used as a test set and a model built from all remaining instances. Though other techniques motivated from the probability and statistics literature can be used to find the best performing model, they in fact return similar results to cross-validation. It is well known that asymptotically leave-one-out validation is identical to Aikake's information criterion (AIC) and that for reasonable (small) values of k that the Bayesian information criterion (BIC) returns similar results to k-fold cross-validation [Moore 2001].

3.3 Unsatisfactory Results

With the Newsgroup data, the actual testing accuracy and their order among four algorithms on each of the six datasets are summarized in Table 1 and 2. As a comparison, the testing accuracy and their order estimated by ten-fold cross-validation are shown in Table 3 and 4, and the corresponding results by leave-one-out are in Table 5 and 6.

DataSet	DT	NB	LR	SVM
Christ-Sales	92.1	87.7	92.0	91.6
Mac-Hardware	81.6	78.9	89.3	76.4
Baseball-Hoc	84.3	75.4	88.6	73.9
MidEast-Elec	85.6	82.8	92.2	78.3
MidEast-Guns	79.7	89.3	89.7	78.6
Auto-Space	85.7	83.2	89.4	79.6

Table 1: Accuracy of Four Classifiers on the Test Set for Various Newsgroup Data Sets

DataSet	1st	2nd	3rd	4th
Christ-Sales	DT	LR	SVM	NB
Mac-Hardware	LR	DT	NB	SVM
Baseball-Hoc	LR	DT	NB	SVM
MidEast-Elec	LR	DT	NB	SVM
MidEast-Guns	LR	NB	DT	SVM
Auto-Space	LR	DT	NB	SVM

Table 2: Accuracy Order of Four Classifiers on Test Set of Various Newsgroup Data Sets

We find that ten-fold cross-validation can be used to accurately predict the order of learner performance most of the time in all but 1 of the 6 data sets (Tables 2 vs 4). As in all our results, an asterisk indicates an incorrect result when compared to the true test set error. However, for leave-one-out validation, in 5 out of 6 data sets, the learner accuracy order is incorrectly predicted (Tables 2 vs 6).

Furthermore, both ten-fold and leave-one-out appear to sometimes provide poor estimates of the learner accuracy (Table 1 vs Tables 3 and 5) with the average difference between the actual error and error estimated by ten-fold (leave-one-out) being 1.6 (3.6) with a minimum of 0 (0) and maximum of 7.5 (14.1).

With biased UCI datasets, we find that both ten-fold and leave-one-out validation do not indicate well which learner performs the best. The actual testing accuracy is summarized in Table 7, and the estimated accuracy by ten-fold cross-validation and leave-one-out are in Tables 8 and 9. If we summarize the results in complete accuracy order, the results would appear pessimistic. Instead, we have chosen a pairwise comparison. For each data set, the four classifiers' accuracy are compared against each other giving rise to $C_4^2/2 = 6$ combinations (DT vs NB, DT vs LR, DT vs SVM, NB vs LR, NB vs SVM and LR vs SVM). Therefore, for our five UCI datasets, there are 30 classifier comparisons (6 per dataset). Table 10 shows the correct pairwise comparison obtained from the test set. Table 11 shows that the results of using ten-fold cross-validation repeated 100 times (at great computation cost) are correct only 17 out of the 30 times. In addition, ten-fold cross-validation is a woeful method to indicate learner accuracy with the average difference being 6.2 (minimum of 0.6 and maximum 20.9) (Tables 7 and 8). The results for leave-one-out validation results (Tables 12) are even worse. For the 30 pairwise comparisons, only 15 have been correctly predicted. Furthermore, the average difference in accuracy is 6.4 with the minimum being 0.4 and the maximum 21.2 (Tables 7 and 9).

The training accuracy results (not shown) are almost identical to the results for leave-one-out validation, and hence is also a poor indicator of the classifiers' true accuracy on the test sets for both the Newsgroup datasets and biased UCI datasets. This is to be accepted

DataSet	DT	NB	LR	SVM
Christ-Sales	91.5	88.1	**91.7**	91.5
Mac-Hardware	85.0	80.0	**89.2**	75.8
Baseball-Hoc	85.7	76.8	**87.7**	73.5
MidEast-Elec	91.5	80.8	**92.2**	75.4
MidEast-Guns	87.2	89.3	**90.2**	78.7
Auto-Space	89.5	84.2	**91.5**	79.7

Table 3: Accuracy for Ten-Fold Cross-Validation of Four Classifiers on Training Set of Various Newsgroup Data Sets. Averaged Accuracy over 100 Trials. c.f. Table 1

DataSet	1st	2nd	3rd	4th
Christ-Sales	*LR	*SVM	*DT	NB
Mac-Hardware	LR	DT	NB	SVM
Baseball-Hoc	LR	DT	NB	SVM
MidEast-Elec	LR	DT	NB	SVM
MidEast-Guns	LR	NB	DT	SVM
Auto-Space	LR	DT	NB	SVM

Table 4: Accuracy Order for Ten-Fold Cross-Validation of Four Classifiers on Training Set of Various Newsgroup Data Sets. Averaged Accuracy over 100 Trials. An "*" indicates a different ordering than Table 2.

DataSet	DT	NB	LR	SVM
Christ-Sales	92.1	87.8	91.9	91.5
Mac-Hardware	85.3	80.6	89.3	75.7
Baseball-Hoc	86.4	76.3	87.5	87.3
MidEast-Elec	92.0	80.5	92.0	92.4
MidEast-Guns	87.8	89.3	90.2	90.2
Auto-Space	89.6	84.1	91.5	91.7

Table 5: Accuracy for Leave-One-Out Validation of Four Classifiers on Training Set of Various Newsgroup Data Sets. c.f. Table 1.

DataSet	1st	2nd	3rd	4th
Christ-Sales	DT	LR	SVM	NB
Mac-Hardware	LR	DT	NB	SVM
Baseball-Hoc	LR	*SVM	*DT	*NB
MidEast-Elec	*SVM	*LR	*DT	*NB
MidEast-Guns	*SVM	*LR	*NB	*DT
Auto-Space	*SVM	*LR	*DT	*NB

Table 6: Accuracy Order for Leave-One-Out Validation of Four Classifiers on Training Set of Various Newsgroup Data Sets. An "*" indicates a different ordering than Table 2.

as the biased training data set is not representative of the unbiased test set.

3.4 An Explanation

Consider the distributions $\mathcal{Q}(\mathbf{x}, y) = \mathcal{Q}(y|\mathbf{x}) \cdot \mathcal{Q}(\mathbf{x})$ from which the test set is drawn and the biased distribution $\mathcal{P}(\mathbf{x}, y) = \mathcal{P}(y|\mathbf{x}) \cdot \mathcal{P}(\mathbf{x})$ from which the training set is drawn. For the feature bias case, which is the focus of this paper, $\mathcal{P}(y|\mathbf{x}) = \mathcal{Q}(y|\mathbf{x})$ but $\mathcal{P}(\mathbf{x}) \neq \mathcal{Q}(\mathbf{x})$. Even if our learner perfectly estimates the true conditional probability $\mathcal{Q}(y|\mathbf{x})$, the estimated generalization error will still most likely be incorrect. Let $\mathcal{P}(y*|\mathbf{x})$ be the probability for the most likely label for a particular instance, then the learner's lowest generalization error possible is $GE = \sum_{\mathbf{x}} \mathcal{Q}(\mathbf{x})(1 - \mathcal{P}(y * |\mathbf{x}))$. However, the lowest generalization error that can be estimated from the training set is $\hat{GE} = \sum_{\mathbf{x}} \mathcal{P}(\mathbf{x})(1 - \mathcal{P}(y * |\mathbf{x})) \neq GE$ as $\mathcal{P}(\mathbf{x}) \neq \mathcal{Q}(\mathbf{x})$. For example in Figure 1:a) the error for Region 1 will be under estimated compared to the region's true error. This is because $\forall \mathbf{x} \in$ Region $1, \mathcal{P}(\mathbf{x}) < \mathcal{Q}(\mathbf{x})$. An over and under occurrence of instances in the training set compared to the test set will lead to systematically under or over stating the generalization error. This is also indicated by our experimental results (Tables 7 and 8). Each and every technique under-estimates the true error for the Breast and Vote data sets, while every technique over-estimates the true error for Iris and Wine. For Pima, three out of the four classification techniques over estimate the true error. Similar results to cross-validation are observed for leave-one-out validation.

4. A NEW APPROACH

The previous experimental results illustrate that traditional cross-validation based approaches cannot be used effectively to determine which learner will outperform the others when the training set is biased. In this section, we propose one that can.

4.1 Basic Idea: ReverseTesting

Assume that θ_a and θ_b are two classifiers trained by algorithms L_a and L_b from the training set D. We are interested to order θ_a and θ_b's accuracy on unlabeled test set T. The intuition is to make use of the testing data's feature vectors but the training data's labels. The conceptual steps of **ReverseTesting** are

1. Classify test data T with θ_a and θ_b. As a result, T_a is the labeled test data by θ_a, and T_b by θ_b.
2. Train "some new classifiers" from T_a and T_b.
3. Evaluate "these new classifiers" on labelled training data D.
4. Based on the accuracy of "these new classifiers" on D, use "some rules" to order the original classifiers' (θ_a and θ_b) accuracy on T.

The name "ReverseTesting" comes from the procedure to "come back" to the training data. In the above basic framework of ReverseTesting, it does not specify either the exact ways to train "new classifiers" or the exact "some rules". We next instantiate these basic procedures with a preferred implementation.

4.2 One Preferred Implementation

The two classifiers, θ_a, θ_b, are constructed by applying learning algorithms L_a and L_b on the training data set D. To determine which one of two classifiers is more accurate on T, the first step is to use both classifiers, θ_a and θ_b, to classify the unlabeled test set to obtain two "labeled" data sets T_a and T_b. In the second step, we construct four new classifiers by applying L_a and L_b on the two labeled test sets, T_a and T_b, respectively, and these four new classifiers are named as θ_a^a, θ_a^b, θ_b^a, and θ_b^b. For example, θ_b^a is the new classifier built using algorithm L_b on T_a or the test set labeled by θ_a. Since the original training set D is labeled, we can use D to evaluate the accuracy of these four new classifiers. Assume that their accuracy on D is $AccTrain_a^a$, $AccTrain_a^b$, $AccTrain_b^a$, and $AccTrain_b^b$ respectively, i.e., $AccTrain_a^b$ is the accuracy of θ_a^b on D. It is important to understand that $AccTrain_a^b$ is **not** the typical training set accuracy, rather it is the accuracy on the training set of a classifier trained by L_a from labeled original test data T_b.

Next, we use two simple rules based on these four accuracy measurements to determine the better performing classifier between θ_a and θ_b on the unlabeled test set T.

DataSets	DT	NB	LR	SVM
Breast	98.9	98.5	98.0	99.0
Iris	92.0	88.0	84.0	66.0
Pima	73.5	72.4	75.0	72.0
Vote	97.0	91.8	97.8	99.3
Wine	55.6	55.6	72.2	66.7

Table 7: Performance of Four Classifiers on Test Set of Various UCI Data Sets

DataSet	DT	NB	LR	SVM
Breast	94.4	95.7	96.6	96.7
Iris	92.9	94	93.9	86.9
Pima	72.6	76.9	77.7	77.3
Vote	95.3	91.2	92.0	94.4
Wine	72.2	75.3	71.6	77.7

Table 8: Accuracy for Ten-Fold Cross-Validation of Four Classifiers on Training Set of Various UCI Data Sets. Averaged Accuracy over 100 Trials

DataSet	DT	NB	LR	SVM
Breast	94.8	95.8	96.8	96.8
Iris	92.4	93.9	94.5	86.4
Pima	69.2	76.8	77.8	77.0
Vote	95.1	90.8	93.2	94.5
Wine	74.6	76.8	71.1	77.5

Table 9: Accuracy for Leave-One-Out Validation of Four Classifiers on Training Set of Various UCI Data Sets.

CONDITION 4.1. *If* $(AccTrain_a^b > AccTrain_a^a) \wedge (AccTrain_b^b > AccTrain_b^a)$*, then* θ_b *is expected to be more accurate than* θ_a *on unlabeled test set* T*.*

CONDITION 4.2. *If* $(AccTrainb_a^a > AccTrain_a^b) \wedge (AccTrain_b^a > AccTrain_b^b)$*, then* θ_a *is expected to be more accurate than* θ_b *on unlabelled test set* T*.*

CONDITION 4.3. *Otherwise,* θ_a *and* θ_b *are tied and hard to distinguish.*

Assume that θ_b is more accurate than θ_a on the testing data T. Under this assumption, there are more examples with correct labels in T_b (or T labeled by θ_b) than T_a. By means of its predicted labels, T_b describes a "concept" that is expected to be closer to the true model than T_a. For a reasonable learning algorithm, the classifier built from T_b is expected to be more accurate than a classifier built from T_a by the same algorithm. Conditions 4.1 and 4.2 capture this reasoning and also rules out that the converse situation since either T_a or T_b is typically a better labeling of the test set.

In summary, if θ_a and θ_b don't have the same accuracy, either i) $(AccTrain_a^a > AccTrain_a^b) \wedge (AccTrain_b^a > AccTrain_b^b)$ when θ_a is more accurate than θ_b, or ii) $(AccTrain_a^b > AccTrain_a^a) \wedge (AccTrain_b^b > AccTrain_b^a)$ when θ_b is more accurate than θ_a, is expected to be true. In other words, if $(AccTrain_a^a > AccTrain_a^b) \wedge (AccTrain_b^a > AccTrain_b^b)$, θ_a is more accurate than θ_b, and if $(AccTrain_a^b > AccTrain_a^a) \wedge (AccTrain_b^b > AccTrain_b^a)$, θ_b is more accurate than θ_a.

When θ_a is more accurate than θ_b, could other orders of accuracy, for example, $(AccTrain_a^a > AccTrain_a^b) \wedge (AccTrain_b^a < AccTrain_b^b)$ be true? In some rare situations, it could happen that a more correctly labeled T may not induce a more accurate classifier. These rare situations include learning algorithms that do not behave reasonably, and those stochastic problems where the true label of some examples have probabilities significantly less than 1 or formally $\exists(\mathbf{x}, y), \mathcal{Q}(y|\mathbf{x}) \ll 1$. When neither Condition 4.1 nor Condition 4.2 is true, θ_a and θ_b are either tied or hard to separate. The complete algorithm is summarized in Figure 3.

Breast	DT	NB	LR	SVM
DT		DT	DT	SVM
NB			NB	SVM
LR				SVM
Iris	**DT**	**NB**	**LR**	**SVM**
DT		DT	DT	DT
NB			NB	NB
LR			LR	LR
Pima	**DT**	**NB**	**LR**	**SVM**
DT		DT	LR	DT
NB			LR	NB
LR				LR
Vote	**DT**	**NB**	**LR**	**SVM**
DT		DT	LR	SVM
NB			LR	SVM
LR				SVM
Wine	**DT**	**NB**	**LR**	**SVM**
DT		DT	LR	SVM
NB			LR	SVM
LR				LR

Table 10: Pairwise Competitive Performance of Four Classifiers on Test Set of Various Biased UCI Data Sets. Each entry indicates which of the classifiers outperformed the other.

4.3 Efficiency

The proposed algorithm is significantly less time consuming than the traditional approaches. With ten-fold cross-validation to compare two learners, we need to build $2 \times 10 \times 100$ models (2 learners and 10 folds repeated 100 times), and with leave one-out validation, $2 \times n$ models where n is the number of instances in the training data set. However, for ReverseTesting, we instead need only to build 6 models (two models from the training set, then four models from the labeled test set). For ℓ models comparison, ten-fold cross-validation and leave-one-out construct $\ell \times 10 \times 100$ and $\ell \times n$ models respectively, and ReverseTesting construct $\ell + 4 \times (\ell + 1) \times \ell/2 = 2\ell^2 + 3\ell$ models. Approximately, only when there were more than 500 algorithms to compare or $\ell > 500$, ReverseTesting could be less efficient than cross-validation.

5. EXPERIMENTAL RESULTS

We begin by evaluating our algorithm on the Newsgroup data sets where ten-fold cross-validation but not leave-one-out validation performed well at choosing the best performing learner. The results are summarized in Table 13. Importantly, we see that for the Newsgroup data sets, which may or may not be biased, that ReverseTesting performs exactly the same as ten-fold cross-validation (Table 4 vs 13)) and significantly better than leave-one-out validation (Table 6). These results are important since Newsgroup datasets have small or no sample selection bias. This illustrates that the proposed algorithm works well when the stationary or non-

Breast	DT	NB	LR	SVM
DT		*NB	*LR	SVM
NB			NB	SVM
LR				SVM
Iris	**DT**	**NB**	**LR**	**SVM**
DT		*NB	*LR	DT
NB			NB	NB
LR				LR
Pima	**DT**	**NB**	**LR**	**SVM**
DT		*NB	LR	*SVM
NB			LR	*SVM
LR				LR
Vote	**DT**	**NB**	**LR**	**SVM**
DT		DT	*DT	*DT
NB			LR	SVM
LR				SVM
Wine	**DT**	**NB**	**LR**	**SVM**
DT		*NB	*DT	SVM
NB			*NB	SVM
LR				*SVM

Table 11: Pairwise Competitive Performance for Ten-Fold Cross-Validation of Four Classifiers on Training Set of Various Biased UCI Data Sets. Averaged Accuracy over 100 Trials. Each entry indicates which of the classifiers outperformed the other. An asterisk means a difference to the correct value in Table 10

Breast	DT	NB	LR	SVM
DT		*NB	*LR	SVM
NB			*LR	SVM
LR				SVM
Iris	**DT**	**NB**	**LR**	**SVM**
DT		*NB	*LR	DT
NB			*LR	NB
LR				LR
Pima	**DT**	**NB**	**LR**	**SVM**
DT		*NB	LR	*SVM
NB			LR	*SVM
LR				LR
Vote	**DT**	**NB**	**LR**	**SVM**
DT		DT	*DT	*DT
NB			LR	SVM
LR				SVM
Wine	**DT**	**NB**	**LR**	**SVM**
DT		*NB	*DT	SVM
NB			*NB	SVM
LR				*SVM

Table 12: Pairwise Competitive Performance for Leave-One-Out Validation of Four Classifiers on Training Set of Various Biased UCI Data Sets. Each entry indicates which of the classifiers outperformed the other. An asterisk means a difference to the correct value in Table 10

biased distribution assumption holds.

For the purposefully biased UCI datasets, the pairwise comparison results of ReverseTesting are shown in Table 14. We see that, out of the 30 comparisons, there are only 5 errors as opposed to 13 errors when using ten-fold cross-validation and 15 errors when using leave-one-out validation. In 3 of the 5 errors, Condition 4.3 occurred and hence no decision on which classifier performed best could be made.

Considering both Newsgroup and UCI datasets, counting the number of *'s or losses in all tables and the total number of all entries (20 for Newsgroup and 30 for biased UCI), the summary is as follows

	#Entry	10-fold	leave-1	RvT
Newsgroup	20	3	15	3
biased UCI	30	13	15	5(3)
Sum	50	16	30	8
% Choose Best Learner		68%	40%	84%

It clearly shows that the proposed algorithm can choose the correct learner most of the time, while ten-fold cross-validation and leave-one-out validation cannot.

6. APPLICATIONS ON DONATION SOLICITATION AND CREDIT CARD FRAUD

We have applied ReverseTesting to two important applications where sample selection bias is known to exist. The first application is charity donation dataset from KDDCUP'98 and the second is a month-by-month data of credit card fraud detection. These problems are particularly interesting since both employ cost-sensitive loss function as opposed to 0-1 loss.

For the donation dataset (Donate), suppose that the cost of requesting a charitable donation from an individual $\mathbf{x}$ is \$0.68, and

DataSet	1st	2nd	3rd	4th
Christ-Sales	*LR	*SVM	*DT	NB
Mac-Hardware	LR	DT	NB	SVM
Baseball-Hoc	LR	DT	NB	SVM
MidEast-Elec	LR	DT	NB	SVM
MidEast-Guns	LR	NB	DT	SVM
Auto-Space	LR	DT	NB	SVM

Table 13: Accuracy Order for ReverseTesting of Four Classifiers on Training Set of Various Newsgroup Data Sets. An "*" indicates a different ordering than Table 2.

the best estimate of the amount that $\mathbf{x}$ will donate is $Y(\mathbf{x})$. Its benefit matrix (converse of loss function) is:

	predict $donate$	predict $\neg donator$
actual $donate$	Y($\mathbf{x}$) - \$.0.68	0
actual $\neg donate$	-\$0.68	0

The accuracy is the total amount of received charity minus the cost of mailing. Assuming that $p(donate|\mathbf{x})$ is the estimated probability that $\mathbf{x}$ is a donor, we will solicit to $\mathbf{x}$ iff $p(donate|\mathbf{x}) \cdot Y(\mathbf{x}) > 0.68$. The data has already been divided into a training set and a test set. The training set consists of 95412 records for which it is known whether or not the person made a donation and how much the donation was. The test set contains 96367 records for which similar donation information was not published until after the KDD'98 competition. We used the standard training/test set splits since it is believed that these are sampled from different individuals thus incuring feature bias [Zadrozny, 2004]. The feature subsets (7 features in total) were based on the KDD'98 winning submission. To estimate the donation amount, we employed the multiple linear regression method. As suggested in [Zadrozny and Elkan, 2001], to

function *ReverseTesting*(L_a, L_b, D, T)
where:

- L_a, L_b are the two learners to compare and choose.
- $D = \{(\mathbf{x}_1, y_1)...(\mathbf{x}_n, y_n)\}$ is the labeled and potentially biased training set.
- $T = \{\mathbf{x}_1...\mathbf{x}_m\}$ is the unlabeled test set.

begin:

1. θ_a and θ_b are the two models trained by applying algorithm L_a and L_b on the training set respectively.
2. T_a is the labeled test set by classifier θ_a. Similarly, T_b is the labeled test set by classifier θ_b.
3. θ_a^b is a classifier trained by applying algorithm L_a on T_b. Similarly, we have θ_a^a, θ_b^a and θ_b^b.
4. Test classifiers $\theta_a^b, \theta_a^a, \theta_b^a$, and θ_b^b on training data D, and their corresponding accuracies on D are denoted as $AccTrain_a^b$, $AccTrain_a^a$, $AccTrain_b^a$, and $AccTrain_b^b$.
5. If $AccTrain_a^b > AccTrain_a^a$ and $AccTrain_b^b > AccTrain_b^a$, then θ_b trained by algorithm L_b is more accurate on unlabeled test data T.
6. Else If $AccTrain_a^a > AccTrain_b^b$ and $AccTrain_b^a > AccTrain_b^b$, then θ_a trained by algorithm L_a is more accurate on T.
7. Otherwise, θ_a and θ_b are either tied or hard to distinguish.

end

Figure 3: A preferred implementation of ReverseTesting to determine the best Learner

avoid over estimation, we only used those contributions between \$0 and \$50.

The second data set is a credit card fraud detection (CCF) problem. Assuming that there is an overhead $v = \$90$ to dispute and investigate a fraud and $y(x)$ is the transaction amount, the following is the benefit matrix:

	predict $fraud$	predict $\neg fraud$
actual $fraud$	$y(x)$ - v	0
actual $\neg fraud$	-v	0

The accuracy is the sum of recovered frauds minus investigation costs. If $p(fraud|\mathbf{x})$ is the probability that $\mathbf{x}$ is a fraud, $fraud$ is the optimal decision iff $p(fraud|\mathbf{x}) \cdot y(x) > v$. The dataset was sampled from a one year period and contains a total of .5M transaction records. The features (20 in total) record the time of the transaction, merchant type, merchant location, and past payment and transaction history summary. We use data of the last month as test data (40038 examples) and data of previous months as training data (406009 examples), thus obviously creating feature bias since no transactions will be repeated.

For cost-sensitive problems, the most suitable methods are those that can output calibrated reliable posterior probabilities [Zadrozny and Elkan, 2001]. For this reason, we use unpruned single decision tree (uDT), single decision tree with curtainlment (cDT), naive Bayes using binning with 10 bins (bNB), and random decision tree with 10 random trees (RDT). Since both datasets are significant in size and leave-one-out would have taken a long time to complete, we only tested 10 cross-validation, repeated for 10 times for each dataset, to compare with ReverseTesting. The detailed accuracy results are summarized in Table 15 and 16, and pairwise orders are summarized in Table 17 and Table 18. ReverseTesting predicted exactly the order as the actual pairwise order on the test datasets for both donation and credit card fraud detection, so its results are

Breast	**DT**	**NB**	**LR**	**SVM**
DT		DT	DT	SVM
NB			*??	SVM
LR				*??
Iris	**DT**	**NB**	**LR**	**SVM**
DT		DT	DT	DT
NB			NB	NB
LR				LR
Pima	**DT**	**NB**	**LR**	**SVM**
DT		DT	*??	DT
NB			LR	NB
LR				LR
Vote	**DT**	**NB**	**LR**	**SVM**
DT		DT	*DT	SVM
NB			LR	SVM
LR				SVM
Wine	**DT**	**NB**	**LR**	**SVM**
DT		DT	*DT	SVM
NB			LR	SVM
LR				LR

Table 14: Pairwise Competitive Performance of Four Classifiers of Various Biased UCI Data Sets using ReverseTesting on Training Set. Each entry indicates which of the classifiers outperformed the other. An entry of "??" indicates that Condition 4.3 occurred and hence no decision could be made. An asterix means a difference to the correct value in Table 10

DataSet	**bNB**	**uDT**	**cDT**	**RDT**
Donate	11334	12577	14424	14567
CCF	412303	537903	691044	712314

Table 15: Accuracy of Four Classifiers on the Test Set for Donation and CCF Data Sets

the same as Table 17. On the other hand, for pairwise order, 10-fold cross validation was correct in 7 out 12 times.

7. INTERPRETATION BASED ON PROBABILITY DISTRIBUTION APPROXIMATION

The proposed algorithm is built around the main idea of Condition 4.1 or "if $(AccTrain_a^b > AccTrain_a^a) \wedge (AccTrain_b^b > AccTrain_b^a)$, then θ_b is expected to be more accurate than θ_a on unlabeled test set T." On the basis of this main idea, we study how ReverseTesting chooses the most accurate classifier to approximate the unbiased distribution $\mathcal{Q}(\mathbf{x}, y)$. We use Kullback-Leilber distance or KL-distance to measure the difference between two distributions.

Recall that testing data set T is drawn from the unbiased joint distribution $\mathcal{Q} = \mathcal{Q}(\mathbf{x}, y)$ (class label y is withheld though), and the training data D is drawn from the biased distribution $\mathcal{P} = \mathcal{P}(\mathbf{x}, y)$. Now assume that $\mathcal{Q}_b = \mathcal{Q}_b(\mathbf{x}, y)$ is the estimate of $\mathcal{Q}(\mathbf{x}, y)$ by classifier θ_b, and similarly, $\mathcal{Q}_a = \mathcal{Q}_a(\mathbf{x}, y)$ by classifier θ_a. We will show that when Condition 4.1 holds true, the KL-distance between $\mathcal{Q}$ and $\mathcal{Q}_b$ is expected to be less than between $\mathcal{Q}$ and $\mathcal{Q}_a$, or $KL(\mathcal{Q}, \mathcal{Q}_b) < KL(\mathcal{Q}, \mathcal{Q}_a)$.

In order to do so, we first show that if Condition 4.1 holds, then

DataSet	bNB	uDT	cDT	RDT
Donate	129	112	135	127
CCF	601434	574123	589416	612434

Table 16: Accuracy for Ten-Fold Cross-Validation of Four Classifiers on Donation and CCF Datasets

Donate	bNB	uDT	cDT	RDT
bNB		uDT	cDT	RDT
uDT			cDT	RDT
cDT				RDT
CCF	**bNB**	**uDT**	**cDT**	**RDT**
bNB		uDT	cDT	RDT
uDT			cDT	RDT
cDT				RDT

Table 17: Pairwise Competitive Performance of Four Classifiers on Testing Data of Donate and CCF. ReverseTesting predicted exactly the same order

the following condition is also true.

OBSERVATION 7.1. *If* $(AccTrain_a^b > AccTrain_a^a) \wedge (AccTrain_b^b > AccTrain_b^a)$ *then,* $KL(\mathcal{P}, \mathcal{Q}_b) < KL(\mathcal{P}, \mathcal{Q}_a)$.

Since both algorithms L_a and L_b were able to compute classifiers θ_a^b and θ_b^b from T_b (whose accuracy on the labeled training data D are $AccTrain_a^b$ and $AccTrain_b^b$ respectively) that are more accurate on the labeled training set D than the other two classifiers θ_a^a and θ_b^a constructed from T_a, then T_b is expected be closer in distribution to D than T_a or $KL(\mathcal{P}, \mathcal{Q}_b) < KL(\mathcal{P}, \mathcal{Q}_a)$. In the unusual situations where this doesn't occur and instead $KL(\mathcal{P}, \mathcal{Q}_b) > KL(\mathcal{P}, \mathcal{Q}_a)$, is when either one or both algorithms L_a and L_b behaves unreasonably and constructs more accurate models from less accurately labeled training set.

OBSERVATION 7.2. *By the definition of accuracy, it is expected to be true that if* θ_b *is more accurate than* θ_a*, then* $KL(\mathcal{Q}, \mathcal{Q}_b) < KL(\mathcal{Q}, \mathcal{Q}_a)$.

Based on the above two observations, we can therefore rewrite Condition 4.1 as:

$$\text{If } KL(\mathcal{P}, \mathcal{Q}_b) < KL(\mathcal{P}, \mathcal{Q}_a) \quad (1)$$
$$\text{then}$$
$$KL(\mathcal{Q}, \mathcal{Q}_b) < KL(\mathcal{Q}, \mathcal{Q}_a)$$

Next, we examine when the above equation holds true. In the simple case, clearly when the stationary distribution assumption holds or $\mathcal{P} = \mathcal{Q}$, the above Eq 1 is trivially correct. This shows that the proposed algorithm should match the probability distribution very well when there is no sample selection bias.

With feature bias, we expand Eq 1 by applying the following set of equations

$$\begin{aligned} \mathcal{P} &= \mathcal{P}(\mathbf{x})\mathcal{Q}(y|\mathbf{x}) = \mathcal{Q}(\mathbf{x})P(s=1|\mathbf{x})\mathcal{Q}(y|\mathbf{x}) \quad (2)\\ \mathcal{Q} &= \mathcal{Q}(\mathbf{x})\mathcal{Q}(y|\mathbf{x}) \\ \mathcal{Q}_a &= \mathcal{Q}_a(\mathbf{x})\mathcal{Q}_a(y|\mathbf{x}) \\ \mathcal{Q}_b &= \mathcal{Q}_b(\mathbf{x})\mathcal{Q}_b(y|\mathbf{x}) \end{aligned}$$

Donate	bNB	uDT	cDT	RDT
bNB		*bNB	cDT	*bNB
uDT			cDT	RDT
cDT				*cDT
CCF	**bNB**	**uDT**	**cDT**	**RDT**
bNB		*bNB	*bNB	RDT
uDT			cDT	RDT
cDT				RDT

Table 18: Pairwise Competitive Performance Predicted by Ten-Fold Cross-Validation on Donate and CCF Data Sets. A * indicates a difference in order from the test data

In the above equations, $Q_a(\mathbf{x}) = Q_b(\mathbf{x}) = \mathcal{P}(\mathbf{x})$ unless θ_a or θ_b is not trained from D directly (such as bootstrap samples) or they are not consistent learners. Using these equations, Eq 1 becomes

$$\begin{aligned} &\text{If } KL(\mathcal{P}(\mathbf{x})\mathcal{Q}(y|\mathbf{x}), \mathcal{P}(\mathbf{x})\mathcal{Q}_b(y|\mathbf{x})) < \quad (3)\\ &\quad KL(\mathcal{P}(\mathbf{x})\mathcal{Q}(y|\mathbf{x}), \mathcal{P}(\mathbf{x})\mathcal{Q}_a(y|\mathbf{x})) \\ &\quad \text{then} \\ &KL(\mathcal{Q}(\mathbf{x})\mathcal{Q}(y|\mathbf{x}), \mathcal{P}(\mathbf{x})\mathcal{Q}_b(y|\mathbf{x})) < \\ &\quad KL(\mathcal{Q}(\mathbf{x})\mathcal{Q}(y|\mathbf{x}), \mathcal{P}(\mathbf{x})\mathcal{Q}_a(y|\mathbf{x})) \end{aligned}$$

The antecedent of Eq 3 can be simplified into $KL(\mathcal{Q}(y|\mathbf{x}), \mathcal{Q}_b(y|\mathbf{x})) < KL(\mathcal{Q}(y|\mathbf{x}), \mathcal{Q}_a(y|\mathbf{x}))$ by removing the constant $P(\mathbf{x})$. In other words, Condition 4.1 is an approximate way to test the precisions of the estimated posterior probability $\mathcal{Q}_a(y|\mathbf{x})$ and $\mathcal{Q}_b(y|\mathbf{x})$ to approximate the true posterior probability $\mathcal{Q}(y|\mathbf{x})$. When $\mathcal{Q}_b(y|\mathbf{x})$ is a better estimate of the posterior probability, the consequent of Eq 3 is expected to be true unless the bias $P(s=1|\mathbf{x})$ is so skewed on a very large portion of the instance space. We ran several tests using synthetic datasets with multiple Boolean variables, and have found that this holds true around 99.9% of the time. Details on this simulation will be in the longer version of this paper.

8. RELATED WORK

The sample selection bias problem has received a great deal of attention in econometrics. There it appears mostly because data are collected through surveys. Very often people that respond to a survey are self-selected, so they do not constitute a random sample of the general population. In Nobel-prize winning work, [Heckman, 1979] has developed a two-step procedure for correcting sample selection bias in linear regression models, which are commonly used in econometrics. The key insight in Heckman's work is that if we can estimate the probability that an observation is selected into the sample, we can use this probability estimate to correct the model. The drawback of his procedure is that it is only applicable to linear regression models. In the statistics literature, the related problem of missing data has been considered extensively [Little and Rubin, 2002]. However, they are generally concerned with cases in which some of the features of an example are missing, and not with cases in which whole examples are missing. The literature in this area distinguishes between different types of missing data mechanisms: missing completely at random (MCAR), missing at random (MAR) and not missing at random (NMAR). Different imputation and weighting methods appropriate for each type of mechanism have been developed. More recently, the sample selection bias problem has begun to receive attention from the machine learning and data mining communities. Fan, Davidson, Zadrozny and Yu [Fan' et al 2005] use the categorization in [Zadrozny, 2004]

to present an improved categorization of the behavior of learning algorithms under sample selection bias (global learners vs. local learners) and analyzes how a number of well-known classifier learning methods are affected by sample selection bias. The improvement over [Zadrozny, 2004] is that the new categorization considers the effects of incorrect modeling assumptions on the behavior of the classifier learner under sample selection bias. In other words, the work relaxes the assumption that the data is drawn from a distribution that could be perfectly fit by the model. The most important conclusion is that most classification learning algorithms could or could not be affected by feature bias. This all depends on if the true model is contained in the model space of the learner or not, which is generally unknown. Smith and Elkan [Smith and Elkan, 2004] provide a systematic characterization of the different types of sample selection bias and examples of real-world situation where they arise. For the characterization, they use a Bayesian network representation that describes the dependence of the selection mechanism on observable and non-observable features and on the class label. They also present an overview of existing learning algorithms from the statistics and econometrics literature that are appropriate for each situation. Finally, Rosset et al. [Rosset et al., 2005] consider the situation where the sample selection bias depends on the true label and present an algorithm based on the method of moments to learn in the presence of this type of bias.

9. CONCLUSION AND FUTURE WORK

Addressing sample selection bias is necessary for data mining in the real world for applications such as merchandise promotion, clinical trial, charity donation, etc. One very important problem is to study the effect of sample selection bias on inductive learners and choose the most accurate classifier under sample selection bias. Some recent works formally and experimentally show that most classifier learners' accuracy could be sensitive to one very common form of sample selection bias, where the chance to select an example into the training set depends on feature vector $\mathbf{x}$ but not directly on class label y. Importantly, this sensitivity depends on whether or not the unknown true model is contained in the model space of the learner, which is generally not known either before or after data mining for real-world applications. This fact makes the problem to choose the most accurate classifier under sample selection bias a critical problem. Our paper provides such a solution.

We first discuss three methods for classifier selection under sample selection bias, ten-fold cross-validation, leave-one-out-validation and structural risk minimization, and empirically evaluate the first two. Our experiments have shown that both the predicted order and value of the learners' accuracy are far from their actual performance on the unbiased test set. In the worst cases, the predicted order is not even better than random guessing. This re-confirms the need to design a new algorithm to select classifiers under sample selection bias.

We propose a new algorithm that is significantly different from those three methods to evaluate a classifier's accuracy. In our problem formulation, we do not assume that the training and testing data are drawn from the same distribution. In other words, they could be drawn from either the same or different distribution. Quite different from the ways of solely relying on labelled training data, we make use of unlabelled test data during model selection. The basic idea and process is to use the competing classifiers trained from the training set to label the test data set, i.e, one labeled test set for each competing classifier, and then re-construct a set of new classifiers from these labeled test sets. We then order the original competing classifiers accuracy based on the new classifiers' accuracy on the training set. The correctness of the proposed algorithm is discussed under accuracy to match true class labels as well as KL-distance to match probability distributions.

Experimental studies have found that when there is little or no sample selection bias, our proposed algorithm predicts the order of performance as good as ten-fold cross-validation and significantly better than leave-one-out validation. Importantly, when there is sample selection bias, our proposed algorithm is significantly better (5 errors including 3 undetermined) than cross-validation (13 errors) and leave-one-out (15) errors among 30 pairwise comparisons. For charity donation solicitation and credit card fraud detection applications where sample bias is a common problem, ReverseTesting is correct in predicting all 12 pairwise orders, while cross-validation is correct in 7 of these cases.

Future Work Although, the algorithm itself does not limit the type of sample selection bias, our paper mainly focuses on feature selection bias. In future work, we will consider other types of sample selection bias. Our algorithm correctly orders the accuracy of competing learners. However, it does not estimate the actual accuracy on the test data itself. This is another challenging problem since the label of the test set is not given in our problem setting. We choose a preferred implementation of ReverseTesting. It is interesting to evaluate other possibilities on how to train the new classifiers from labeled test data and the rules to induce performance order.

10. REFERENCES

Fan W., Davidson I., Zadrozny B. and Yu P., (2005), An Improved Categorization of Classifier's Sensitivity on Sample Selection Bias, 5th IEEE International Conference on Data Mining, ICDM 2005.

Heckman, J. (1979). Sample selection bias as a specification error. *Econometrica*, 47:153–161.

Little, R. and Rubin, D. (2002). *Statistical Analysis with Missing Data*. Wiley, 2nd edition.

McCallum, A. (1998). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. CMU TR.

Moore, A. *A Tutorial on the VC Dimension for Characterizing Classifiers*, Available from the Website: www.cs.cmu.edu/˜awm/tutorials

Rennie, J. *20 Newsgroups*, (2003). Technical Report, Dept C.S., MIT.

Rosset, S., Zhu, J., Zou, H., and Hastie, T. (2005). A method for inferring label sampling mechanisms in semi-supervised learning. In *Advances in Neural Information Processing Systems 17*, pages 1161–1168. MIT Press.

Shawe-Taylor J., Bartlett P., Williamson R., Anthony M., (1996), "A Framework for Structural Risk Minimisation" Proceedings of the 9th Annual Conference on Computational Learning Theory.

Smith, A. and Elkan, C. (2004). A bayesian network framework for reject inference. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 286–295.

Vapnik, V., *The Nature of Statistical Learning*, Springer, 1995.

Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proceedings of the 21th International Conference on Machine Learning*.

B. Zadrozny and C. Elkan. (2001). Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the Seventh ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD01)*.