

# On Sample Selection Bias and Its Efficient Correction via Model Averaging and Unlabeled Examples

Wei Fan\*

Ian Davidson†

## Abstract

Sample selection bias is a common problem encountered when using data mining algorithms for many real-world applications. Traditionally, it is assumed that training and test data are sampled from the same probability distribution, the so called “stationary or non-biased distribution assumption.” However, this assumption is often violated in reality. Typical examples include marketing solicitation, fraud detection, drug testing, loan approval, school enrollment, etc. For these applications the only labeled data available for training is a biased representation, in various ways, of the future data on which the inductive model will predict. Intuitively, some examples sampled frequently into the training data may actually be infrequent in the testing data, and vice versa. When this happens, an inductive model constructed from biased training set may not be as accurate on unbiased testing data if there had not been any selection bias in the training data. In this paper, we first improve and clarify a previously proposed categorization of sample selection bias. In particular, we show that unless under very restricted conditions, sample selection bias is a common problem for many real-world situations. We then analyze various effects of sample selection bias on inductive modeling, in particular, how the “true” conditional probability  $P(y|\mathbf{x})$  to be modeled by inductive learners can be misrepresented in the biased training data, that subsequently misleads a learning algorithm. To solve inaccuracy problems due to sample selection bias, we explore how to use model averaging of (1) conditional probabilities  $P(y|\mathbf{x})$ , (2) feature probabilities  $P(\mathbf{x})$ , and (3) joint probabilities,  $P(\mathbf{x}, y)$ , to reduce the influence of sample selection bias on model accuracy. In particular, we explore on how to use unlabeled data in a semi-supervised learning framework to improve the accuracy of descriptive models constructed from biased training samples.

## 1 Introduction and Motivation

In the past few years, there has been active research on sample selection bias, as shown in the number of papers published on this topic (Nobel Prize winning work [Heckman, 1979] and more recent works [Little and Rubin, 2002, Zadrozny, 2004, Smith and Elkan, 2004, Rosset et al., 2005, Chawla and Karakoulas, 2005, Fan’ et al 2005, Fan and Davidson, 2006, Davidson and Fan, 2006]. A brief survey can be found in Section 7. This topic is important for data mining applications since traditionally many algorithms make the “stationary or non-biased distribution assumption” that each and every training and test instance is draw identically and independently from a common distribution  $P(\mathbf{x}, y)$  [Vapnik, 1995]. However, this is rarely the case in practice. For many applications, the training data set follows a distribution,  $Q(\mathbf{x}, y)$ , that is different from  $P(\mathbf{x}, y)$  [Fan and Davidson, 2006]. An accurate model trained from and evaluated on distribution,  $Q(\mathbf{x}, y)$ , may not be as accurate on distribution  $P(\mathbf{x}, y)$ . Consider the drug testing application 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 [Zadrozny, 2004]. 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.

There are several important and inter-related problems that involve data mining under sample se-

\*IBM T.J.Watson Research Center, Hawthorne, NY 10532, weifan@us.ibm.com

†Department of Computer Science, University at Albany, State University of New York, Albany, NY 12222, davidson@cs.albany.edu

lection bias. The work in [Zadrozny, 2004] and later [Fan’ et al 2005] discuss four types of sample bias and a taxonomy to categorize learners into two types, those that are effected by some types of biases (global learners) and those that are not (local learners). In [Zadrozny, 2004], Zadrozny proposes an efficient method to correct one particular type of sample selection bias (feature bias as discussed in Section 2). However, it requires firstly knowing that a training set is biased and a formal model to quantify the distribution of this sampling bias in order to learn the correction. Other earlier work [Fan and Davidson, 2006] illustrates that cross-validation is a poor method of evaluating the best classifier when the training set is biased, and proposes an alternative method that effectively uses unlabeled and unbiased data to order classifiers’ accuracy. The focus of this paper is to correct sample bias without the extra requirements imposed by the previous method [Zadrozny, 2004], instead we use various forms of model averaging and unlabeled data. The advantage of our proposed method is that it does not need to know either the exact type of sample selection bias or a probability distribution that formally models the bias. It is important to note that we do not claim that ensemble techniques in general can be used to help address sample bias, in fact the opposite is true of bagging and boosting [Davidson and Fan, 2006].

## 2 Sample Selection Bias and Its Practical Impact

Following the notations initially introduced in [Zadrozny, 2004], assume that  $s$  is a random variable which takes either 1 or 0 (“sampled” or “not sampled”). Then, the predicate “ $s = 1$ ” denotes that a labeled example  $(\mathbf{x}, y)$  (where  $\mathbf{x} = (x_1, x_2, \dots, x_k)$  is a feature vector of  $k$  feature values, and  $y$  is a class label) is sampled into the training set  $\mathcal{T}$  from the universe of all labeled examples  $\mathcal{D}$ , and “ $s = 0$ ” denotes that  $(\mathbf{x}, y)$  is not selected. Additionally, the joint probability  $P(\mathbf{x}, y) = P(y|\mathbf{x}) \cdot P(\mathbf{x})$  denotes the *unbiased* or *true probability* to observe example  $(\mathbf{x}, y)$  in the universe of examples  $\mathcal{D}$ . Many inductive learning algorithms model  $P(\mathbf{x}, y)$  either via approximating conditional probability  $P(y|\mathbf{x})$  or both  $P(y|\mathbf{x})$  and  $P(\mathbf{x})$  in the same time. Since the inductive model is trained from labeled training instances, it is important to examine if these training instances are an accurate representation of the target concept to be modeled. This is exactly where sample selection bias ought to be considered.

With these definitions, sample selection bias is best described by a conditional dependency of  $s = 1$  (or the predicate that “ $(\mathbf{x}, y)$  is sampled” to be true) on both  $\mathbf{x}$  and  $y$ , or formally,  $P(s = 1|\mathbf{x}, y)$ . As initially proposed

in [Zadrozny, 2004], there can be four types of “explicit” conditional dependencies:

- *No sample selection bias* or  $P(s = 1|\mathbf{x}, y) = P(s = 1)$ . In other words, the selection process is independent from both feature vector  $\mathbf{x}$  and class label  $y$ .
- *Feature bias* or  $P(s = 1|\mathbf{x}, y) = P(s = 1|\mathbf{x})$ . The selection process is conditionally independent from  $y$  given  $\mathbf{x}$ . It is important to understand that feature bias does not imply  $s = 1$  is completely independent from  $y$ .
- *Class bias* or  $P(s = 1|\mathbf{x}, y) = P(s = 1|y)$ . The selection process is conditionally independent from  $\mathbf{x}$  given  $y$ . Similarly, it does not imply that  $s = 1$  is completely independent from  $\mathbf{x}$ .
- *Complete bias* or  $P(s = 1|\mathbf{x}, y)$  The selection process is dependent on both  $\mathbf{x}$  and  $y$ .

To understand the implications of this previously proposed categorization, some further analyses and explanations are necessary. In particular, it is important to formally define how the dependency probability  $P(s = 1|\mathbf{x}, y)$  is calculated, as it is open to interpretation when initially introduced in [Zadrozny, 2004].

There could be two different ways on how to interpret  $P(s = 1|\mathbf{x}, y)$ . The practical interpretation is as follows.

**DEFINITION 2.1.** *For a given or particular training set  $\mathcal{T}$ ,  $P(s = 1|\mathbf{x}, y)$  (or abbreviated as  $P(s|\mathbf{x}, y)$ ) is the probability for a labeled example  $(\mathbf{x}, y)$  to be selected from the universe of examples  $\mathcal{D}$  into this training set  $\mathcal{T}$ .*

Under this definition, obviously, if a labeled example  $(\mathbf{x}, y)$  does not appear at all in the training set  $\mathcal{T}$ , then  $P(s|\mathbf{x}, y) = 0$ . For those examples sampled into  $\mathcal{T}$ , let  $N(\mathbf{x}, y)$  denote the number of times that  $(\mathbf{x}, y)$  appear in the training data, then  $P(s|\mathbf{x}, y) \propto N(\mathbf{x}, y)$ . The normalization denominator is not critical.

Yet, a different, but less practical, interpretation of  $P(s|\mathbf{x}, y)$  is to consider the fact that *different* training data sets of the same size (as compared to a particular training set, used in the above definition) can be sampled from the universe of labeled examples. Given this consideration,  $P(s|\mathbf{x}, y)$  is the probability to select  $(\mathbf{x}, y)$  from  $\mathcal{D}$ , when multiple training sets of the same size are sampled *exhaustively*. This interpretation is quite different from Definition 2.1, since it effectively assumes that training set can be infinitely sampled.

In this paper and for future works, we rely on and recommend the first interpretation or Definition 2.1

due to several practical reasons. In reality, we are normally given just “one particular training data set”, and the task of inductive learning is to construct an accurate model for the universe of examples  $\mathcal{D}$  based on this possibly biased training set  $\mathcal{T}$ . With sample selection bias  $P(s|\mathbf{x}, y)$  formally defined, the training set distribution is thus,  $Q(\mathbf{x}, y) = P(s = 1, \mathbf{x}, y) = P(s = 1|\mathbf{x}, y)P(\mathbf{x}, y)$ . Next, we analyze some important properties of sample selection bias and its effects on  $Q(\mathbf{x}, y)$  under Definition 2.1.

Most importantly, we demonstrate that under Definition 2.1, sample selection bias is a “ubiquitous” problem, in the sense that either sample selection bias is difficult to avoid in practice or the conditional probability represented in the biased data,  $P(y|\mathbf{x}, s)$ , can potentially be quite different from the true unbiased conditional probability  $P(y|\mathbf{x})$ . Later in the paper, we shall provide two frameworks that can effectively overcome the influence of sample selection bias and increase overall accuracy.

**2.1 Sample Selection Bias in Practice** Consider the definition of “no sample selection bias case” or  $P(s|\mathbf{x}, y) = P(s)$ . It means that the selection process is conditionally independent from both  $\mathbf{x}$  and  $y$ . However, under Definition 2.1 of  $P(s|\mathbf{x}, y)$ , this is rather difficult to guarantee. For example, if the training set  $\mathcal{T}$  does not include every possible feature vector  $\mathbf{x}$ , there is at least a dependency of  $s = 1$  on  $\mathbf{x}$ . Then, it is more appropriate to consider the sampling process as either feature bias or complete selection bias. Indeed, it is rather difficult to exhaustively collect every feature vector for any real-world applications, “no sample selection bias” is likely an unrealistic assumption in practice.

**2.2 Misrepresented Conditional Probability** Since many inductive learning algorithms approximate the conditional probability of the unknown true function,  $P(y|\mathbf{x})$ , in different ways, it is important to examine how this “true” quantity might have been misrepresented in the training data due to various types of sample selection bias. Formally, the conditional probability represented in the training dataset is  $P(y|\mathbf{x}, s)$ , but not  $P(y|\mathbf{x})$ . We next discuss the relationship between  $P(y|\mathbf{x}, s)$  and  $P(y|\mathbf{x})$ , given different type of sample selection bias under Definition 2.1.

**2.2.1 Feature Bias** By definition of conditional probability,  $P(s|\mathbf{x}, y) = P(s|\mathbf{x})$  is the same as  $\frac{P(s, \mathbf{x}, y)}{P(\mathbf{x}, y)} = \frac{P(s, \mathbf{x})}{P(\mathbf{x})}$ . By re-arranging  $P(\mathbf{x})$  and  $P(s, \mathbf{x})$ , it becomes  $\frac{P(s, \mathbf{x}, y)}{P(s, \mathbf{x})} = \frac{P(\mathbf{x}, y)}{P(\mathbf{x})}$  or  $P(y|\mathbf{x}, s) = P(y|\mathbf{x})$ . One could easily conclude that feature bias does not change condi-

tional probability. However, this derivation is problematic due to two reasons. First, for feature bias, strictly speaking,  $P(y|\mathbf{x}, s)$  is undefined for any feature vector  $\mathbf{x}$  not sampled into the training set. But, in the derivation, it is implicitly assumed that  $P(s, \mathbf{x}) \neq 0$ . In other words, the derivation is only valid for sampled feature vectors. Secondly,  $P(y|\mathbf{x}, s) = P(y|\mathbf{x})$  is true when  $y$  given  $\mathbf{x}$  is deterministic that is  $P(y|\mathbf{x}) = \{0, 1\}$ . However, for stochastic examples, where  $P(y|\mathbf{x}) < 1$  for all class labels, in general,  $P(y|\mathbf{x}, s)$  approximates  $P(y|\mathbf{x})$  only when  $\mathbf{x}$  is sampled “exhaustively.” And indeed, the derivation to show  $P(y|\mathbf{x}, s) = P(y|\mathbf{x})$  implicitly assumes exhaustive sampling. For example, assume that  $P(y = +|\mathbf{x}) = 0.8$  for a given feature vector  $\mathbf{x}$ . If the training data contains only 3 examples with feature vector  $\mathbf{x}$ , then the best approximation that can be estimated from this training set is either  $P(y = +|\mathbf{x}, s) = 0.67 = \frac{2}{3}$  or  $P(y = +|\mathbf{x}, s) = 1 = \frac{3}{3}$ .

As a summary of the above discussion, the correct assumption about conditional probability given feature bias is

**ASSUMPTION 2.1.** *Given feature bias,  $P(y|\mathbf{x}, s = 1) = P(y|\mathbf{x}, s)$  is “approximately” correct for a large number of sampled examples. That is, the MSE between  $P(y|\mathbf{x}, s)$  and  $P(y|\mathbf{x})$  for sampled examples is expected to be reasonably small.*

**2.2.2 Class Bias and Complete Bias** For class bias and complete bias, it is impossible to derive  $P(y|\mathbf{x}, s) = P(y|\mathbf{x})$  without making further assumptions (such as the problem is deterministic or the bias can actually be reduced into feature bias anyhow, etc). Then, it is realistic to assume that:

**ASSUMPTION 2.2.** *Given class bias and complete bias,  $P(y|\mathbf{x}, s) \neq P(y|\mathbf{x})$  for most examples. That is, the MSE between  $P(y|\mathbf{x}, s)$  and  $P(y|\mathbf{x})$  is expected to be large.*

**2.2.3 Examples** Diagrammatically, the effect of feature and complete sample bias on the decision regions can be seen in Figure 1. The figure shows how the quantity of the true conditional probability  $P(+|\mathbf{x})$  changes over the instance space. In the center of the instance space, the probability of an instance being positive is the largest. As can be seen for the feature bias case (the top left plot), if some instances are not present or rare in the training set, then the empirical probability estimate could be incorrect at least for that region. Additionally, as shown in the bottom left plot, as complete bias over-samples or under-sample some regions in the instance space as well as under-samples the positive class towards the center, the estimated probability by

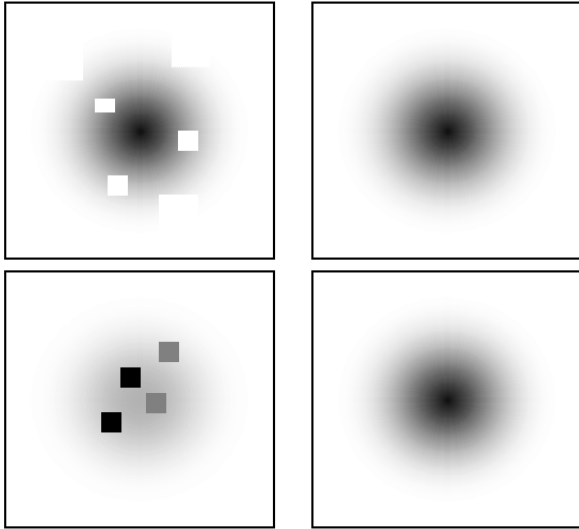


Figure 1: Misrepresented Concept due to Sample Selection Bias. The two plots on the right are the unbiased testing sets. There are two classes and the shades illustrate the change of the true conditional probability  $P(+|\mathbf{x})$  over the instance space, thus, those in the center are more likely to be positive. The two plots on the left demonstrate how “feature bias” (top left) and “complete bias” (bottom left) might have changed the true concept or conditional probability represented in the training data

most learners on the universe of examples could be far from being correct.

### 3 Effect of Sample Selection Bias on Inductive Model

Although, it is shown in [Fan and Davidson, 2006] that when the inductive model happens to be the “true model” that generates the data, then “feature bias” does not significantly affect the algorithm that builds this model, one cannot generalize this conclusion to class bias or complete bias. The main argument for feature bias is that when the model happens to be the correct model,  $P(y|\mathbf{x}, \theta) = P(y|\mathbf{x})$  by definition. In which,  $\theta$  is the inductive model and  $P(y|\mathbf{x}, \theta)$  denotes the non-trivial dependency of the estimated probability on the choice of model  $\theta$ . Since  $P(y|\mathbf{x}, s) = P(y|\mathbf{x})$  is approximately correct given feature bias, it is trivially true that  $P(y|\mathbf{x}, s) = P(y|\mathbf{x}, \theta)$ . However, given class bias and complete bias,  $P(y|\mathbf{x}) \neq P(y|\mathbf{x}, s)$  in general.

Most importantly, for a new application, we rarely know if a chosen model happens to be the correct model that generates the data. Exploring the “correct model”

usually requires domain knowledge that attempts to understand the hidden mechanism behind labeled data, and this is usually not the focus of black-box data mining algorithms. The importance of this work is to show that sample selection bias, regardless of its type, is expected to affect most learning algorithms and most applications. On the other hand, different mining algorithms are likely affected differently by the same sample selection bias. Thus, their differences can be taken advantage of to minimize the effect of sample selection bias. The focus of the rest of the paper is on how to correct the effect of sample selection bias in order to build highly accurate models.

## 4 Correcting Sample Selection Bias

Consider a model space from which the best model must be estimated but from biased training data. Typically, this best model may be highly probable with respect to the biased training data, but will perform poorly on the unbiased test set. However, different models that are built to fit biased training sets are likely to have large variations in their performance on the unbiased test set. The question is how we can reduce these variations. It is known that model averaging techniques and their variations are useful variance reduction techniques, as we will formally show, that can be used to correct the influence of sample selection bias.

Given these observations, we propose two major frameworks and their variations to correct the influence of sample selection bias. The choice to select which framework depends on the assumption of the relative number of examples with reasonable conditional probabilities,  $P(y|\mathbf{x}, s) = P(y|\mathbf{x})$ , or Assumptions 2.1 and 2.2. The first framework is based on average of estimated conditional probabilities by multiple inductive models, and it is suitable when this number is large, such as feature bias. On the other hand, the second framework is suitable when this number is small. In summary, this alternative estimates “joint probability”  $P(\mathbf{x}, y)$  via a mixture of conditional probability models  $P(y|\mathbf{x})$  and feature probability models  $P(\mathbf{x})$ . In addition, we explore semi-supervised learning concept to use “unbiased and unlabeled” examples to improve the accuracy of models originally built from biased examples.

**4.1 Conditional Probability Averaging** Consider an example  $\mathbf{x}$  whose true probability distribution for class label  $y$  is  $P(y|\mathbf{x})$ . We have  $T$  models,  $\theta_1$  to  $\theta_T$ , that approximate this true probability differently. We use “model averaging” of these  $T$  models to minimize the “expected error” to approximate the true probability by

any single model.

$$(4.1) \quad CA(\mathbf{x}) = E_{P(\theta)}(P(y|\mathbf{x}, \theta))$$

The expected MSE for a given model  $\theta_k$  to approximate the true probability over the universe of examples is simply the expected difference between the true and estimated probabilities squared, then integrated over the joint probability of all possible examples.

$$(4.2) \quad \begin{aligned} Error_{\theta_k} &= \Sigma_{\mathbf{x}, y} P(\mathbf{x}, y) (P(y|\mathbf{x}) - P(y|\mathbf{x}, \theta_k))^2 \\ &= E_{P(\mathbf{x}, y)} [P(y|\mathbf{x})^2 - 2P(y|\mathbf{x})P(y|\mathbf{x}, \theta_k) + \\ &\quad P(y|\mathbf{x}, \theta_k)^2] \end{aligned}$$

The expected MSE if a single model is chosen at random from any of the  $T$  models (this reflects the fact we normally do not know which model is the most accurate if they were evaluated exhaustively on the universe of examples) is then the same as Eq 4.2 except for that there is an additional term  $P(\theta)$  in the expectation.

$$(4.3) \quad \begin{aligned} Error_{SingleModel} &= \Sigma_{\theta_k} \Sigma_{\mathbf{x}, y} P(\mathbf{x}, y) \times \\ &\quad (P(y|\mathbf{x}) - P(y|\mathbf{x}, \theta_k))^2 \\ &= E_{P(\theta), P(\mathbf{x}, y)} [P(y|\mathbf{x})^2 - 2P(y|\mathbf{x})P(y|\mathbf{x}, \theta_k) + \\ &\quad P(y|\mathbf{x}, \theta_k)^2] \end{aligned}$$

Note in the above, we did not average the model's predictions, we only analyzed the average performance of a single model chosen at random from the  $T$ . Next, we analyze how we can improve the expected performance by averaging these  $T$  models.

If we were to average the predictions of  $T$  model, then their expected performance is

$$(4.4) \quad \begin{aligned} Err_{CA} &= \Sigma_{\mathbf{x}, y} P(\mathbf{x}, y) (P(y|\mathbf{x}) - E_{P(\theta)}[P(y|\mathbf{x}, \theta)])^2 \\ &= E_{P(\mathbf{x}, y)} [P(y|\mathbf{x})^2 - 2P(y|\mathbf{x})E_{P(\theta)}[P(y|\mathbf{x}, \theta)] + \\ &\quad E_{P(\theta)}[P(y|\mathbf{x}, \theta)]^2] \\ &\leq E_{P(\mathbf{x}, y)} [P(y|\mathbf{x})^2 - 2P(y|\mathbf{x})E_{P(\theta)}[P(y|\mathbf{x}, \theta)] + \\ &\quad E_{P(\theta)}[P(y|\mathbf{x}, \theta)]^2] \\ &\leq Error_{SingleModel} \end{aligned}$$

$$\text{as } E[f(x)]^2 \leq E[f(x)^2]$$

Note the the differences between  $E_{P(\theta)}[P(y|\mathbf{x}, \theta)]^2$  and  $E_{P(\theta)}[P(y|\mathbf{x}, \theta)^2]$ . Therefore, over the universe of examples  $P(\mathbf{x}, y)$ , conditional probability averaging against

any single model, on average, would perform better than a single model.

However, conditional probability averaging is limited by the performance of single models, and its performance could not be significantly better if a majority of single models are far off from the true probabilities. Considering different forms of sample selection bias, conditional probability averaging is more appropriate for feature bias rather than class bias and complete bias, where there are expected to be a larger number of examples in the training data with approximately correct  $P(y|\mathbf{x})$ , thus resulting in reasonably accurate single models.

**4.2 Joint Probability Averaging** Classification can be made by either joint probability  $P(\mathbf{x}, y)$  or conditional probability  $P(y|\mathbf{x})$ . But they do not have to be estimated from the same models, thus, their predictions can be different. Next, we discuss methods to build joint probability  $P(\mathbf{x}, y)$  models that are able correct large number of inaccurate conditional probability models that averaging can not significantly improve.

The intuition is that joint probability  $P(\mathbf{x}, y)$  can be decomposed into the product of two terms,  $P(y, \mathbf{x}) = P(y|\mathbf{x}) \cdot P(\mathbf{x})$ . We can construct different models to estimate conditional probability  $P(y|\mathbf{x})$  and "feature probability"  $P(\mathbf{x})$ , and then use model averaging to combine different models,

$$(4.5) \quad JA(\mathbf{x}) = \sum P(y|\mathbf{x}, \theta_k^c) \cdot P(\mathbf{x}|\theta_k^f)$$

in which,  $\theta_k^c$  is the conditional probability model indexed at  $k$  and  $\theta_k^f$  is the corresponding feature model.

Before discussing technical details, we provide an example to show how joint probability models can correct sample selection bias that conditional probability averaging cannot. Assume that we have two conditional probability models and their estimates for feature vector  $\mathbf{x}$  is  $P_1(+|\mathbf{x}) = 0.8$  or  $P_1(-|\mathbf{x}) = 0.2$  as well as  $P_2(+|\mathbf{x}) = 0.4$  or  $P_2(-|\mathbf{x}) = 0.6$ , respectively. Obviously, there is a discrepancy between these two models. Conditional probability averaging will predict class +, since the averaged probability is 0.6. However, the difference between these two models, one favors - over + more than the other model, could be taken advantage of, if we instead train two separate feature models  $P_1(\mathbf{x})$  and  $P_2(\mathbf{x})$  and use the product rule to estimate the joint probability. To be precise, assume that  $P_1(\mathbf{x}) = 0.05$  and  $P_2(\mathbf{x}) = 0.4$ . Then the joint probability estimate is  $P(+, \mathbf{x}) = 0.05 \times 0.8 + 0.4 \times 0.4 = 0.2$  and  $P(-, \mathbf{x}) = 0.05 \times 0.2 + 0.4 \times 0.6 = 0.25$ . Therefore, prediction now becomes - instead of +.

There could be many different models to estimate  $P(y|\mathbf{x})$  and  $P(\mathbf{x})$ , and different ways to combine them to

calculate the joint probability. However, in order for this framework to make a different prediction, under 0-1 loss, from conditional probability averaging, two conditions have to be met.

1. There need to be differences among conditional probability models, i.e., some models favor one class while others favor a different class. Otherwise, if the most likely label estimated by every model is the same, both joint probability and conditional probability averaging will always predict the same class.
2. Conditional probability  $P(y|\mathbf{x}, \theta_i)$  for the correct class and the associated feature probability  $P(\mathbf{x}|\theta_j)$  ought to be positively correlated to overcome those inaccurate  $P(y|\mathbf{x}, \theta)$ . In other words,  $P(y|\mathbf{x}, \theta_i) \cdot P(\mathbf{x}|\theta_j)$  need to dominate the equation.

### 4.3 Using Unlabeled and Unbiased Examples

If unlabeled and unbiased examples are available, they can be used to improve both  $P(\mathbf{x})$  and  $P(y|\mathbf{x})$  models.

Clearly, since class label is not considered,  $P(\mathbf{x})$  models can be built from both labeled and unlabeled examples. A decision tree based implementation is discussed in Section 5.

If the chosen learning algorithm to build conditional probability model is descriptive (rather than generative), such as decision trees, one could use unlabeled data to improve  $P(y|\mathbf{x})$  directly from unlabeled data. The idea is to use estimated  $P(y|\mathbf{x})$  to label unlabeled data, then include the labeled data into the training data and re-construct  $P(y|\mathbf{x})$  models. The new models are expected to be quite different from the old ones, since both the structure and the set of parameters within the new structure is new for descriptive methods. This procedure stops when the estimated probability for unlabeled example remains statistically stable. A similar approach called “semi-supervised discriminant learning” has been previously proposed in [Vittaut et al., 2002]. For feature bias, this process is expected to be helpful at least for 0-1 loss problems, since the proof given in [Vittaut et al., 2002] is applicable. The advantage of this semi-supervised process is to directly incorporate those  $\mathbf{x}$ 's, previously excluded by sample selection bias, into the models.

The above semi-supervised process is not directly applicable to generative models, such as logistic regression. The reason is that the form of the hypothesis is pre-fixed, usually as an equation, and learning estimates value of parameters within the equation. When the same equation is used to predict on unlabeled examples and then use the prediction to re-compute parameter values. the new set of parameter values are not

expected to be significantly different from the original values. The prediction by new models won't be significantly different.

## 5 Descriptive Implementation in Decision Trees

Based on the frameworks discussed in Sections 4.1, 4.2 and 4.3, we propose a straightforward implementation using ensemble of decision trees. There are several reasons to choose decision tree over other methods.

- It is able to estimate both conditional probability and feature probability.
- Decision tree is inherently descriptive, which is necessary to use unlabeled examples to improve conditional probability model.
- There are many known methods to generate multiple decision trees from the same dataset.

**5.1 Probability Calculation** To use a decision tree to compute conditional probability,  $P(y|\mathbf{x})$ , we simply divide the number of examples belonging to class  $y$  by the total number of labeled examples in the classifying leaf node. For example there are 10 examples in the leaf node and 2 of them are of class +, then  $P(+|\mathbf{x}) = 0.2$ .

On the other hand, to estimate feature probability  $P(\mathbf{x})$ , we divide the total number of examples in the classifying leaf node by the total number of examples sorted by the tree. For example, the total number of training examples that generates the decision tree is 1000, and the classifying leaf node has 10 examples,  $P(\mathbf{x}) = \frac{10}{1000} = 0.01$ .

Additionally, it is straightforward to use unlabeled examples to improve  $P(\mathbf{x})$  estimation by decision trees. To do so, we simply “classify” those unlabeled examples by the decision tree. But at each leaf node, we maintain a separate counter that records the number of “unlabeled” examples sorted into this leaf node. For the same leaf node discussed in the previous paragraph, assume that 2 of the 500 unlabeled examples are sorted into this leaf node. Then, the improved feature probability, by taking into account both labeled and unlabeled examples, is  $\frac{10+2}{1000+500} = 0.008$ .

**5.2 Adaption into the Frameworks** Both conditional probability averaging (Section 4.1) and semi-supervised descriptive learning (Section 4.3) are directly applicable on decision trees without much adaptation. However, for joint probability averaging (Section 4.2), we examine two alternative implementations. In the first implementation, we use the *same decision tree* to estimate both conditional and feature probabilities, and

then multiply them together. Formally,

$$(5.6) \quad \sum_k P(y|\mathbf{x}, \theta_k)P(\mathbf{x}|\theta_k) = \sum_k P(\mathbf{x}, y|\theta_k)$$

Obviously,  $P(y|\theta_k)$  and  $P(\mathbf{x}|\theta_k)$  are strongly correlated as they are being predicted from the same decision tree. As discussed in Section 4.2, it is likely to make different predictions from conditional probability averaging, especially when unlabeled data examples could be used to improve  $P(\mathbf{x}|\theta_k)$ .

A different implementation combines “uncorrelated” conditional probability model and feature probability models from different trees or

$$(5.7) \quad \sum_k [P(y|\mathbf{x}, \theta_k) \times \sum_{j \neq k} P(\mathbf{x}|\theta_j)]$$

We call it *cross* joint probability averaging, and Eq 5.6 *self* joint probability averaging. Our intuition is to examine the contribution from different amount of correlations in constructing an accurate model.

Since one appropriate baseline comparison is a single decision tree, we construct multiple decision trees as follows. Each tree is constructed from the original labeled training set. In order to create different trees, we randomly sample a subset of features at each leaf node, and then choose the one with the highest information gain among all features in the feature subset. This is similar to Random Forest, however the tree is constructed from original training set but not bootstraps. When unlabeled data is given, it is populated into each tree, and each node of the tree maintains counters for both labeled and unlabeled examples.

## 6 Experiments

The main focus of experimental studies is to empirically evaluate different methods’ accuracy as different types of sample selection bias are introduced into the same problem. For each type of sample selection across different datasets, it is useful to find out if there are any “robust” methods that return the best overall results across these different problems.

**6.1 Experimental Setup** We have chosen a mixture of problems, varying in, number of classes, number of examples, types and number of features. Five datasets are from UCI, and CAR4 is a parametric query optimization dataset. Their prior class distribution are summarized in Table 1. In particular, CART4 is generated from a query optimizer and query plan is known to be deterministic according to selectivity measures.

Table 1: Data Set Characteristics

	Prior Class Distribution
Adult	2 classes and 24% positive
SJ	3 classes with distribution (52%, 24%, 24%)
SS	3 classes with distribution (20%, 55%, 25%)
Pendig	10 classes, with relatively even distribution
ArtiChar	10 classes with even distribution
CAR4	4 classes with (57%, 37%, 4%, 2%)

We generate different types of sample selection bias using the original training data. Details on bias simulation can be found in the following sections. In the case of semi-supervised learning (to improve either  $P(\mathbf{x})$  models or  $P(y|\mathbf{x})$  models), we use the entire testing data (ignoring its labels though) as the unbiased examples. For decision tree ensemble, 10 trees are constructed. To construct each tree, 80% of features at each leaf node is randomly chosen to form the feature subset.

**6.2 Generate Feature Bias** This is to directly follow the definition of “feature bias”  $P(s = 1|\mathbf{x}, y) = P(s = 1|\mathbf{x})$ . In other words, the selection process is directly dependent on feature vector  $\mathbf{x}$ , and conditionally independent from class label  $y$  given  $\mathbf{x}$ . To do so, we sort the dataset according to each feature (dictionary sort for categorical feature and numerical sort for continuous feature), and remove the top 25% examples from the sorted list, thus creating a bias on each sorted feature. The results on each dataset is summarized in Table 2, where each number is the average of tests biased on different features. The abbreviations used in the table are as follows:

- CA is conditional probability averaging, as defined in Eq 4.1.
- sCA is semi-supervised descriptive learning applied on CA. In other words, unlabeled test data are used to improve conditional probability models in an iterative procedure, described in Section 4.3.
- JA is joint probability averaging, described in Eq 5.6, where both conditional probability and feature probability are estimated by the same decision tree, but in different ways.
- sJA refers to unlabeled testing data used to improve feature probability  $P(\mathbf{x})$  only for each tree in JA or Eq 5.6. The process to describe how to use unlabeled examples to improve  $P(\mathbf{x})$  can be found in paragraph 3 of Section 5.1. It is important to understand that unlabeled testing data

cannot be used to improve both  $P(\mathbf{x})$  and  $P(y|\mathbf{x})$  “separately.” After the semi-supervised descriptive learning labels the unlabeled data, they would be already built into the model at the next iteration.

- JAx is “cross” joint probability averaging, described by Eq 5.7, where conditional probability and feature probability are estimated by the “different” decision trees .
- sJAx refers to the use of unlabeled data to improve feature probability  $P(\mathbf{x})$  (but not conditional probability) n Eq 5.7.

In Table 2, the highest accuracy is highlighted in bold font. Clearly, any of the ensemble methods, conditional probability averaging, joint probability averaging, either using or not using unlabeled data, have obtained significantly higher accuracy (from 2% to 4%) than the single decision tree C4.5, implying different ensemble methods can overcome feature bias. The difference among ensembles is not significant, although CA or conditional probability averaging and JAx or cross joint probability average, appear to be slightly more accurate than JA or self joint probability averaging. On average, unlabeled data have insignificant effect to improve multiple models under feature bias.

As an additional experiment, we compare the results of the same method applied on two datasets with the same size, one with bias and one without bias. A method is considered less sensitive to the bias if the difference between these two results are trivial. To do so, we create such a data set with  $m$  Boolean features giving an instance space  $X = (\mathbf{x})$  of size  $2^m$ , and  $m$  ranges from five to ten. Then for each data instance or feature vector, we generate its class label either “+” or “-” according to the following conditional probability distribution,  $Q(y = “+”|\mathbf{x}) = \sum_i I(x_i = “T”)/m$  where  $I(\dots)$  is an indicator function, that is proportional to the number of features with the value “T”. In the unbiased test set of two hundred instances, each instance is equally likely. To generate the biased training set of another two hundred instances, approximately 25% of instances in the entire instance space are chosen at random at each test run, and their probability to be sampled into the trained set is reduced by 25%. In effect, this reduction increase the probability of the other instances to be selected into the training sets. This selection bias is only related to the feature vector  $\mathbf{x}$  and independent from the class label  $y$ . We have found that the difference in accuracy for C4.5 is approximately 7%, but each of the averaging method’s difference is  $< 1\%$ .

**6.3 Generate Class Bias** Class bias, or  $P(s = 1|\mathbf{x}, y) = P(s = 1|y)$  describes the situation that

the training data has different prior class probability distribution from the true probability distribution. To simulate different ways of class bias, we partition the original training dataset into multiple “class bins”, where each class bin keeps all the examples belonging to that particular class. To generate a biased training set, we first randomly choose a prior probability distribution for each class, fix the training set size to be the same as the original unbiased training set, then sample examples from each class bin with replacement. For a two class problem, if the randomly generated prior class probability is (0.2, 0.8) and the training set size is 100, then 20 examples will be sampled from the positive class and 80 examples will be sampled from the the negative class. We repeat the experiment 100 times to cover different class biased distributions for each dataset and the average results of 100 runs are summarized in Table 3.

Comparing Table 3 with Table 2, the effect of class bias on models’ accuracy is more significant than feature bias. For the adult dataset, C4.5’s accuracy drops from 81.5% on feature bias down to 72.5% on class bias. This is obviously due to effect of class bias that reduces the number of examples with  $P(y|\mathbf{x}, s) = P(y|\mathbf{x})$ . Between single decision tree and each of the ensemble methods, the reduction in accuracy by ensemble methods is much less. For the adult dataset, C4.5 drops 9%, while each of the ensembles drops up to 5%. Among all methods, 4 out of 6 most accurate models are sJA or self joint probability averaging updated with unlabeled examples. It is particularly interesting to compare the results between JA and sJA. On average, sJA increases the accuracy of JA from 2% to 5%. It clearly shows that under class bias, unlabeled data is useful to correct inaccurate  $P(y|\mathbf{x}, \theta)$  models with the self joint probability averaging framework. The accuracy of JAx or cross joint probability averaging is either the highest or very close to the highest for all six datasets. In addition, unlabeled data doesn’t appear to have any significant effect on JAx, which can be observed by comparing the results of JAx and sJAx.

Since the amount of class bias can be easily quantified, it is useful to study the correlation between the amount of bias and each method’s relative performance. The amount of bias can be quantified as  $B(y) = \sum_{y \in \mathcal{Y}} |P(y|s) - P(y)|$ . In order to compare across different sets, we record the maximum  $B(y)$  of all 100 test runs, and use it to normalize each test run’s  $B(y)$ . Between two method  $M_1$  and  $M_2$ , such as between sJA and JA, we record the the difference in accuracy, such as sJA - JA, and the respective amount of normalized bias. To compare across different datasets, the difference in accuracy is also normalized. Between

any two methods, 600 pairs of correlation can be obtained. We are particularly interested in comparing CA with C4.5, as well as sJA with JA. The results are plotted in Figure 2. The plot on the left is for CA vs. C4.5 and the one on the right is for JA vs sJA. Both plots show a linear-like correlation, which is to be accepted that as more class bias is introduced, the benefit to use multiple models (CA in the left plot) over the single model becomes more obvious, and the advantage to use unlabeled examples to improve JA also becomes rather clear.

**6.4 Generate Complete Bias** Complete bias chooses examples according to both the feature vector and class labels in the same time. In other words, some labeled examples  $(\mathbf{x}, y)$  are more frequent than others and vice versa in the biased sample, that carries a different distribution from the unbiased distribution. Since it is hard to justify the utility of any single mapping function to generate complete bias across different datasets, bootstrap sampling, or multiple samples without replacement, is an effective method to cover many different ways to generate complete bias. Inside one bootstrap sample, some examples  $(\mathbf{x}, y)$  are sampled more often than others and some examples in the training set are not sampled at all. It is important to point out that to use bootstrap samples for complete bias evaluation, each algorithm is trained on one bootstrap then evaluated on the testing data. This is different from bagging where multiple bootstraps are used to generate multiple models using the same inductive learner, and then multiple models are combined by voting.

The results are summarized in Table 4. Comparing with Tables 2 and 3, the effect of complete bias on reduction of accuracy is between feature bias and class bias. In other words, the accuracy of each model is roughly between the respective results of feature bias and class bias. Complete bias does not modify prior class probability distribution like class bias, but it changes conditional probability by over-sampling some labeled examples and under-sampling others. Similar to earlier results, ensemble-based methods apparently can achieve higher accuracy than any of the single models, and the increase in accuracy is from 1% to 5%. The difference among ensemble methods are up to about 2.5%. Similar to feature bias but different from class bias, the best performing ensembles are not one method consistently, but alternative among CA, sCA, JA, sJA, JAx, and sJAx. Unlabeled data help improve the original models in 14 of 18 experiments. In particular, for sJA. unlabeled labels appear to improve  $P(\mathbf{x})$  models consistently that return higher accuracy than JA for each dataset.

**6.5 Real-World Datasets** We obtained two datasets whose training data and testing data follow different distributions. The first one is the KDDCUP'98 donation dataset. The chances that someone who has donated recently will donate again in the new campaign is very low. In this aspect, the data is likely biased on both feature vector and class label. Another interesting aspect of this dataset is that it is a cost-sensitive problem, in which, the cost-function is to maximize the received donation minus the cost to mail solicitation letters to potential donors. The second dataset is a credit card fraud detection dataset. Ignoring the time stamp of the transaction, the chances that very similar transactions happen again in a short period of time is very low. The dataset is obviously an example of feature bias. In addition, credit card fraud detection is also cost-sensitive that learning is to maximize the amount of recovered fraud minus the overhead of fraud investigation. In both cases, normalized probability is used in order to make the final prediction, either  $P(y|\mathbf{x}) \cdot \text{profit} \geq \text{cost}$  of doing business or replace with  $P(y, \mathbf{x})$  for joint probability models. The detail of each dataset can be found in [Fan and Davidson, 2006]. Results with various methods proposed in this paper are summarized in Table 5. Similar to earlier results with simulated bias, the advantage of using the ensemble to improve a single model is clearly demonstrated with an increase in total profits from 20% to 30%.

## 7 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).

Table 2: Results on Datasets with Feature Bias

Data Set	C4.5	CA	sCA	JA	sJA	JAx	sJAx
Adult	81.2	84.3	<b>84.7</b>	83.2	82.4	84.5	84.4
SJ	89.2	<b>92.7</b>	91.9	92.1	92.3	92.6	<b>92.7</b>
SS	67.3	69.2	<b>70.1</b>	68.7	68.5	68.9	68.7
Pendig	89.3	93.4	93.7	92.7	93.1	<b>93.9</b>	93.8
ArtiChar	69.3	<b>71.7</b>	71.6	71.3	71.4	71.5	71.3
CAR4	91.7	95.4	95.5	94.7	95.1	<b>95.6</b>	95.2

Table 3: Results on Datasets with Class Bias

Data Set	C4.5	CA	sCA	JA	sJA	JAx	sJAx
Adult	72.5	80.2	79.2	79.4	<b>82.1</b>	81.7	81.5
SJ	71.4	77.4	76.3	73.4	77.7	78.5	<b>78.9</b>
SS	53.5	63.7	62.1	59.4	<b>64.5</b>	62.7	62.4
Pendig	70.4	78.7	77.9	74.5	79.1	<b>79.2</b>	78.9
ArtiChar	51.4	59.2	60.1	58.2	<b>62.3</b>	61.2	61.5
CAR4	69.5	84.4	83.9	84.2	<b>89.2</b>	88.4	87.9

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.

There is much work that makes use of unlabeled data to improve classification accuracy such as [Peng et. al 2003]. However, in that work, as well as in the vast majority of data mining, the basic assumption is that the training set (both labeled and unlabeled) are drawn from the same distribution as the test set. Recently in [Huang et al., 2006], an approach to correct sample bias by re-weighting training examples has been proposed. It estimates a ratio between the unbiased distribution and biased training distribution; using our notations introduced in Sections 1 and 2,  $\beta(\mathbf{x}, y) = \frac{P(\mathbf{x}, y)}{Q(\mathbf{x}, y)}$  is estimated. Under their assumption that “the conditional probability represented in the training data is the same as the unbiased true distribution” (recall that this is a stronger assumption than assuming “Feature Bias” as analyzed in Section 2.2), this can be reduced as the ratio of feature distribution, i.e.,  $\beta(\mathbf{x}, y) = \frac{P(y|\mathbf{x})P(\mathbf{x})}{Q(y|\mathbf{x})Q(\mathbf{x})} = \frac{P(\mathbf{x})}{Q(\mathbf{x})}$ . Thus, with “unlabeled” data, it can be estimated per feature vector  $\mathbf{x}$ . Further assumptions and simplifications are necessary in order to estimate this ratio, particularly when some feature vectors do not appear in one of the datasets or there are continuous features. With this ratio, a learning algorithm can concentrate more on those feature vectors appearing more often in the “unlabeled and unbiased” data. When the assumptions made by this proposal

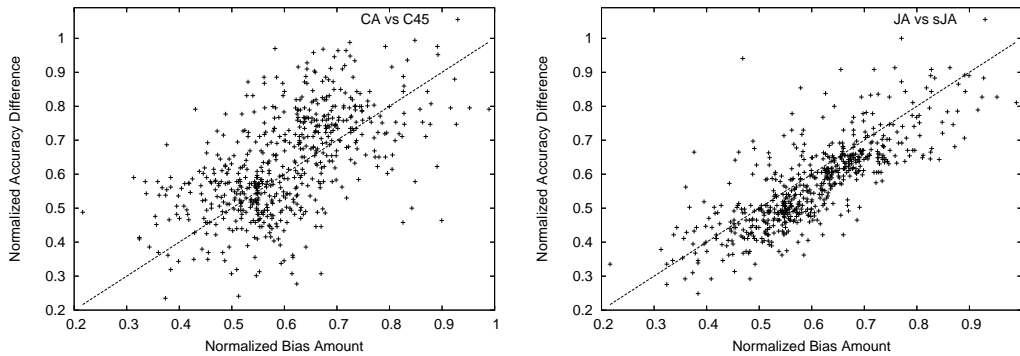


Figure 2: Correlation between amount of bias and performance

Table 4: Results on Datasets with Complete Bias

Data Set	C4.5	CA	sCA	JA	sJA	JAx	sJAx
Adult	79.3	<b>82.4</b>	81.9	81.4	81.7	82.2	81.4
SJ	78.4	81.5	82.2	83.1	<b>84.2</b>	<b>84.2</b>	84.1
SS	61.4	64.3	<b>65.2</b>	63.9	64.2	64.1	64.9
Pendig	82.1	82.9	<b>84.5</b>	83.4	83.7	83.9	84.2
ArtiChar	67.5	70.4	<b>70.5</b>	69.8	70.4	69.7	69.9
CAR4	88.9	90.4	91.2	89.7	90.2	<b>91.6</b>	91.4

are true (conditional probability in the biased training data is correct, labeled training examples are sufficiently large to cover the feature space, and etc.) as well as the chosen algorithm is sensitive to re-weighting, it is expected to be effective.

In [Zadrozny, 2004], an approach to correct “feature bias” is proposed by re-sampling. However, the assumption is to know the probability of feature bias, in other words,  $P(s = 1|\mathbf{x})$  is assumed to be known in advance. However, we do not assume to know this, but only have “unlabeled and unbiased” data. This is a different assumption, thus, the two approaches are not directly comparable. The method in [Huang et al., 2006] could be regarded as an improvement over [Zadrozny, 2004] that it estimates  $P(s = 1|\mathbf{x})$  from both labeled and unlabeled data under the assumption that “conditional probability is not changed by feature bias.”

In [Davidson and Fan, 2006], they discussed a number of situations, such as, small training sets, feature bias, class label noise, and many label problems, where bagging and boosting perform poorly, but efficient model averaging of conditional probability estimators perform well. The major difference of this paper from [Davidson and Fan, 2006] is that we first formally define how the sample selection bias probability  $P(s = 1|\mathbf{x}, y)$  should be calculated practically, and how true conditional probability may have been misrepre-

sented by different types of sample selection bias. In terms of algorithm design, we propose new methods that use a mixture of models to estimate the joint probability distribution as well as semi-supervised learning procedures to use unlabeled examples to correct sample selection bias.

## 8 Conclusion

We have elaborated on four previously proposed basic types of sample selection bias. We have shown that under a practical definition of sample selection bias (Definition 2.1), many datasets in real-world applications are potentially biased. Importantly, we emphasize that one of the most important effects of any sample selection bias that concerns data mining practitioners is the relative number of examples whose class label conditional probability can be correctly estimated by a model as a result of sample selection bias. We then analyze how the different types of sample selection bias can affect this relative number. In summary, feature bias is expected not to affect this number as significantly as class bias and complete bias.

Given these analyses, we extend two frameworks based on model averaging to correct the effect of every type of sample selection bias. The first framework averages several uncorrelated conditional probability estimates, while the second mixes conditional probability

Table 5: Results (\$profit - cost) on Two Real-World Problems that are Biased

Data Set	C4.5	CA	sCA	JA	sJA	JAx	sJAx
Donation	12577	14911	14923	15212	<b>15322</b>	15102	15075
Credit Card	696506	8410310	841120	<b>844530</b>	843928	841120	840940

estimates with feature probability estimates in order to compute the joint probability. The accuracy of these frameworks is then further improved with unlabeled and unbiased examples in a semi-supervised learning framework. As an improvement over earlier methods to correct sample selection bias, the proposed approaches do not assume an exact type of bias and does not assume a formal model to quantify the distribution of the bias in order to make a correction.

With six datasets, we have generated every type of sample selection bias. For every and each type of sample selection bias, the proposed model averaging methods have achieved 2 to 20% higher accuracy than a single model. In particular, when the training data has class bias, joint probability averaging models (JAx) consistently perform better than any other methods. In most situations, unlabeled data used in a semi-supervised framework to improve  $P(\mathbf{x})$  and  $P(y|\mathbf{x})$  models can further increase the accuracy by 1 to 2%.

For data mining practitioners to solve sample selection bias problems in real-world applications, we recommend a procedure as follows. First, it is useful to approximately know the type of sample selection. For class bias, cross joint probability models (JAx) are expected to return the best overall results. Otherwise, for feature bias and complete bias, model averaging of conditional probability models are usually highly accurate. When the type of bias is difficult to guess, any proposed methods in this paper are expected to perform better than a single model. In addition, unlabeled examples have been demonstrated to further increase the accuracy of these new frameworks.

It is important to note that we are not claiming that ensemble techniques in general can correct sample selection bias. It is previously shown [Davidson and Fan, 2006] that boosting and bagging perform poorly when there is feature selection bias.

## References

[Fan' et al 2005] 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.  
 [Fan and Davidson, 2006] Fan W., and Davidson I., (2006) ReverseTesting: An Efficient Framework to Select

Amongst Classifiers under Sample Selection Bias, 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2006.  
 [Davidson and Fan, 2006] Davidson I., and Fan W., (2006) When Efficient Model Averaging Out-Performs Boosting and Bagging, 17th European Conference on Machine Learning and 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD06).  
 [Heckman, 1979] Heckman, J. (1979). Sample selection bias as a specification error. *Econometrica*, 47:153–161.  
 [Little and Rubin, 2002] Little, R. and Rubin, D. (2002). *Statistical Analysis with Missing Data*. Wiley, 2nd edition.  
 [Peng et. al 2003] Peng, K., Vucetic, S., Han, B., Xie H. and Obradovic, Z. (2003). Exploiting Unlabeled Data for Improving Accuracy of Predictive Data Mining Proc. Third IEEE Int'l Conf. Data Mining, Melbourne, FL, pp. 267-274.  
 [Chawla and Karakoulas, 2005] Chawla N. V. and Karakoulas G. (2005). Learning From Labeled And Unlabeled Data: An Empirical Study Across Techniques And Domains Journal of Artificial Intelligence and Research, Volume 23, pages 331-366.  
 [Huang et al., 2006] Huang, J., Smola, A., Gretton, A., Borgwardt, K., and Scholkopf, B. (2006). Correcting Sample Selection Bias by Unlabeled Data In *Proceedings of Neural Information Processing System Conference (NIPS'2006)*, December 4-6, 2006, Vancouver, BC, Canada.  
 [Rosset et al., 2005] 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.  
 [Smith and Elkan, 2004] 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, 1995] Vapnik, V.. (1995). *The Nature of Statistical Learning*, Springer, 1995.  
 [Vittaut et al., 2002] Vittaut, J., Amini, M., and Gallinari, P. (2002), Learning Classification with Both Labeled and Unlabeled Data. In *ECML 2002*, pages 468-479.  
 [Zadrozny, 2004] Zadrozny, B. (2004). Learning and evaluating classifiers under sample selection bias. In *Proceedings of the 21th International Conference on Machine Learning*.