Type Independent Correction of Sample Selection Bias via Structural Discovery and Re-balancing

Jiangtao Ren[*]   Xiaoxiao Shi[†]   Wei Fan[‡]   Philip S. Yu[§]

Abstract

Sample selection bias is a common problem in many real world applications, where training data are obtained under realistic constraints that make them follow a different distribution from the future testing data. For example, in the application of hospital clinical studies, it is common practice to build models from the eligible volunteers as the training data, and then apply the model to the entire populations. Because these volunteers are usually not selected at random, the training set may not be drawn from the same distribution as the test set. Thus, such a dataset suffers from "sample selection bias" or "covariate shift". In the past few years, much work has been proposed to reduce sample selection bias, mainly by statically matching the distribution between training set and test set. But in this paper, we do not explore the different distributions directly. Instead, we propose to discover the natural structure of the target distribution, by which different types of sample selection biases can be evidently observed and then be reduced by generating a new sample set from the structure. In particular, unlabeled data are involved in the new sample set to enhance the ability to minimize sample selection bias. One main advantage of the proposed approach is that it can correct all types of sample selection biases, while most of the previously proposed approaches are designed for some specific types of biases. In experimental studies, we simulate all 3 types of sample selection biases on 17 different classification problems, thus 17×3 biased datasets are used to test the performance of the proposed algorithm. The baseline models include decision tree, naive Bayes, nearest neighbor, and logistic regression. Across all combinations, the increase in accuracy over non-corrected sample set is 30% on average using each baseline model.

Keywords: Sample Selection Bias, Structure, Sampling, Clustering

[*]Department of Computer Science, Sun Yat-Sen University, Guangzhou, China. issrjt@mail.sysu.edu.cn. The author is supported by the National Natural Science Foundation of China under Grant No. 60703119

[†]Department of Computer Science, Sun Yat-Sen University, Guangzhou, China. isshxx@mail.sysu.edu.cn.

[‡]IBM T.J.Watson Research, USA. weifan@us.ibm.com

[§]University of Illinois at Chicago, USA. psyu@cs.uic.edu

1 Introduction

Many machine learning algorithms assume that the training data follow the same distribution as the test data on which the model will later be used to make predictions. However, in real world application, training data are often obtained under realistic conditions, which may easily cause a different training and test distribution. Formally, the training dataset follows a distribution of $Q(\mathbf{x}, y)$, whereas the test set is dominated by a distribution $P(\mathbf{x}, y) \neq Q(\mathbf{x}, y)$. For example, in the application of credit risk assessment, it is often the case that different banks build their assessment models based on their own customers, and then the models are applied to the new clients. But because the new customers may not follow the same distribution as their previous ones, imprecise assessment can happen from time to time. This could be one of the reasons of New Century's "sub prime mortgage crisis and credit crunching". To be specific, such a dataset is said to suffer from "covariate shift" or "sample selection bias", owing to the different distribution between training set and test set. In the past few years, aiming to correct sample selection bias, an increasing number of methods have been proposed, as shown in the number of papers published on this topic (Nobel Prize winning work [7] and more recent works [1]∼[6] and [11]∼[14]). In essence, much of these previous work reduce sample selection bias by exploring the proportion of $\frac{P(\mathbf{x},y)}{Q(\mathbf{x},y)}$. A brief review can be found in Section 6.

On the other hand, as shown in [5], if the structure of a dataset as function of $P(\mathbf{x}, y)$ is known in advance, for example, a parametric model, one could use it to avoid sample selection bias directly. However, it is often not the case that the structure is known apriori. In this paper, we propose a clustering based method to explore the intrinsic structure of $P(\mathbf{x}, y)$. Intuitively, the complete dataset is composed of several data groups. Within each group, the examples $(\mathbf{x}, y)$ are similar. For example, in a hospital clinical study, the eligible volunteers can be partitioned into two groups, one of which may consist of the people who are more likely to suffer from "drug allergy" while the others may be less likely. Suppose that we build a learning model just from some of the groups, e.g., only from the "less likely allergic group". The learning model has high possibility to be imprecise, and even does harm to some of the population when it is put into practice because the "more likely allergic

group" has been neglected due to sample bias. A "universal" learning model should be built on the samples obtained from all of the groups. Nonetheless, in order to reflect the true distribution of the whole dataset, the samples should be drawn under the same proportion in each group, without bias or emphasis/neglect on anyone. It is intuitively true that this new sample set should be unbiased. We provide an illustration in Figure 1 as well as a formal proof in Lemma 3.1.

We propose an approach on the basis of this intuition. In our method, clustering is chosen to discover the structural data groups of $P(\mathbf{x}, y)$, by which sample selection bias can be revealed if the distribution of samples in different groups is imbalanced. Furthermore, with the explored structure, a new sample set with a uniform distribution can be generated by evenly selecting samples from each cluster under the same proportion. It is important to note that some cluster may have so few or no labeled data at all that some unlabeled data in the cluster have to be drawn into the new sample set, so as to balance the distribution. In our algorithm, these unlabeled data will be assigned the most likely labels according to a criterion formally proven in Lemma 3.2.

In summary, we propose a method that can effectively reduce sample selection bias and significantly increase the accuracy of classifier. It has the following key properties:

- *Type independent*: Our approach is aimed to correct sample selection bias based on the structure of a dataset. It is independent of any exact type of sample selection bias, which is different from most of the previous works (See Section 3.2 and Section 6 for more details).

- *Straightforward*: Since automatic clustering that estimates the optimal number of clusters is utilized to discover the structure of a dataset $P(\mathbf{x}, y)$, our approach is easy and effective to put into practice and can evidently reduce sample selection bias.

Empirical studies can be found in Section 5, in which we simulate all 3 types of sample selection biases on 17 different classification problems. Then, up to 51 biased datasets are used to test the proposed algorithm. In addition, the new sample set, generated by the proposed method, will be evaluated on different classifiers, and the increase in accuracy is around 30% on average.

2 Sample Selection Bias

Assume that the event $s = 1$ denotes that a labeled instance $(\mathbf{x}, y)$ is selected from the domain D of instances into the training set L, and that $s = 0$ denotes that $(\mathbf{x}, y)$ is not chosen. When constructing a classification model, we only have access to instances where $s = 1$. In [14] and later [5], four different types of sample selection biases are clearly discussed according to the dependency of s on $\mathbf{x}$ and y:
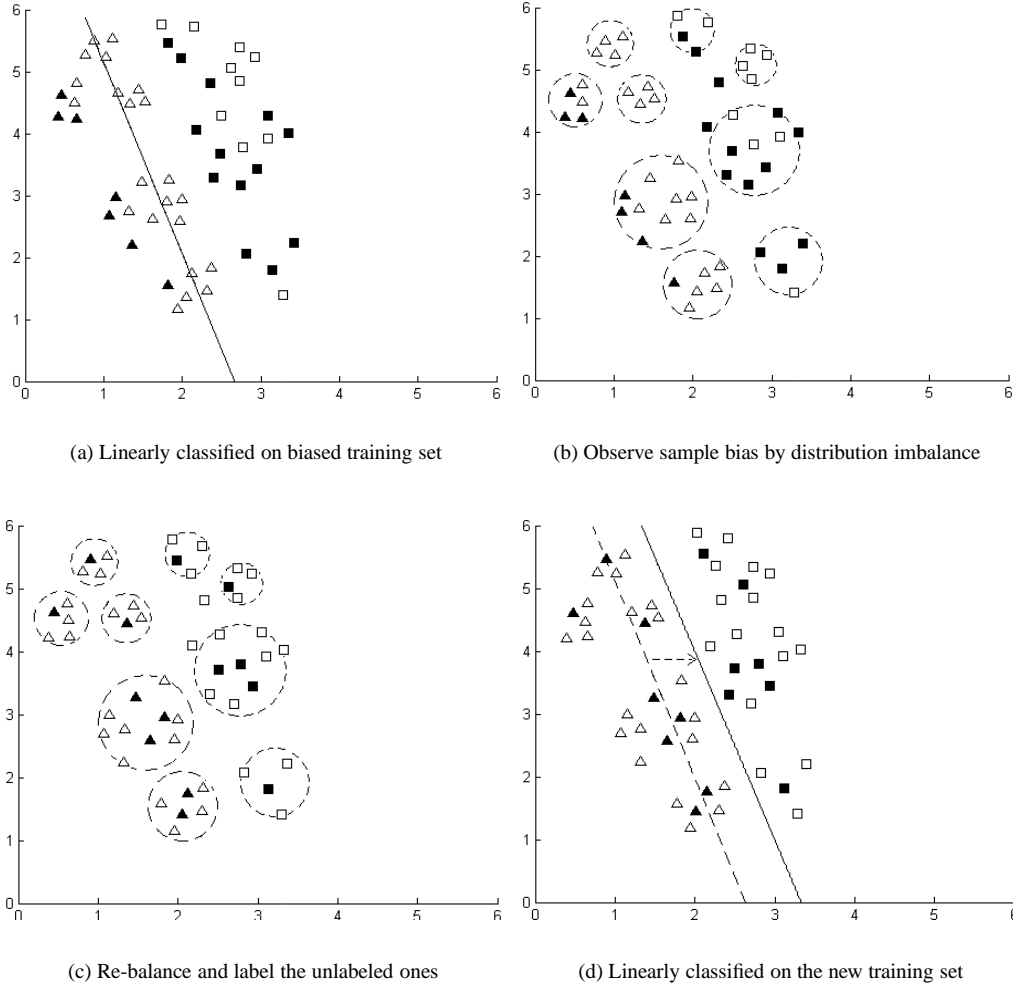
Table 1: Definition of symbols

Sym.	Definition
$\mathbf{X}$	Instance Space
$\mathbf{x_i}$	Instance(without label)
	$\mathbf{X} = \{\mathbf{x_1}, \cdots, \mathbf{x_i}, \cdots\}$
y_i	The class label of $\mathbf{x_i}$
U	Unlabeled dataset
$\mathbf{u}$	A unlabeled data, $\mathbf{u} \in U$
L	Labeled dataset
$\mathbf{l}$	A labeled data, $\mathbf{l} \in L$
L^+	Labeled dataset with class label "+"
L^-	Labeled dataset with class label "−"
α	the proportion of labeled data.
	$P(s) = \frac{|L|}{|\mathbf{X}|} = \alpha$
K	The number of clusters
C_i	The ith cluster
$\mathbf{c_i}$	The center of the ith cluster
	$\mathbf{c_i} = \frac{1}{|C_i|} \sum_{\mathbf{x_j} \in C_i} \mathbf{x_j}$

Note: to simplify the description, only binary class classification problem is discussed based on Table 1. But the proposed method can also work well on multi-class problems, which can be found in the experiments in Section 4.

1. No sample selection bias or s is independent from both $\mathbf{x}$ and y, which is $P(s|\mathbf{x}, y) = P(s)$. In other words, the selection variable s is a random variable completely independent from both the feature vector $\mathbf{x}$ and the true class label y.

2. Feature bias or s is independent of y given $\mathbf{x}$, which is $P(s|\mathbf{x}, y) = P(s|\mathbf{x})$. In [5], this kind of sample selection bias is said to be "naturally exists" since it widely happens in real world application.

3. Class bias or s is only independent of $\mathbf{x}$ given y, which is $P(s|\mathbf{x}, y) = P(s|y)$. It means that the selected sample is biased on the label y.

4. Complete bias, there is no assumption about any independence of s give $\mathbf{x}$ and y, which is also said to be "arbitrary bias" or "missing not at random" (MNAR).

3 Structural Discovery and Re-balancing

In this section, we will first give an overview of the proposed method, including the basic intuition, as well as the framework of the algorithm. Further technical details about the algorithm will be discussed in Section 3.2 and Section 3.3. Table 1 indicates some symbols that will be used in the rest of the paper. In order to make the following description more convenient to be understood, only bi-class classification problem will be discussed in this section. But the algorithm can also work well on multi-class problems in a similar way, as described in Section 4.

(a) Linearly classified on biased training set

(b) Observe sample bias by distribution imbalance

(c) Re-balance and label the unlabeled ones

(d) Linearly classified on the new training set

Note: There are two classes in the dataset, "△" and "□";
The selected samples of "△" are depicted as "▲", while those selected of "□" are depicted as "■".

Figure 1: Sample selection bias correction. In Figure (a), a linear classifier is trained on the sample set and half of the △ are assigned to the wrong class; Then a clustering algorithm is performed on the dataset, the result of which is depicted in Figure (b), in which sample bias can be clearly observed by the sample imbalance in different clusters. In Figure (c), we uniformly obtain samples from each cluster, and mark each of the unlabeled sample with its nearest labeled neighbor. The classification result of the new training set is depicted in Figure(d).

3.1 Basic Intuition Sample selection bias can happen from time to time in real world practice. In order to reduce sample bias, we use clustering to discover the structure of a dataset $P(\mathbf{x}, y)$. In our proposed method, we assume and also show formally in Lemma 3.1 that the distribution of samples in a biased dataset is likely to be different from one cluster to another. In other words, the collected samples may be distributed more in some of the clusters than others. This is obviously sample distribution imbalance. Ideally, to make the distribution of samples more uniform or balanced, we should evenly obtain samples from each cluster. Thus,

the bias ought to be reduced in the newly selected sample set since it is uniformly sampled.

We illustrate the idea in Figure 1. All of the sub-figures in Figure 1 represent a dataset which contains only two classes indicated as '△' and '□' respectively. With a biased sample set, a linear classifier is trained to classify the two classes, as shown in Figure 1(a). We can observe that almost half of the '△' data are assigned to the wrong class. To explore the structure of the dataset by similarity, a clustering algorithm is performed, and the result is plotted in Figure 1(b), with a dotted circle to indicate one cluster.

In Figure 1(b), an obvious imbalance of the distribution of samples can be observed in different clusters: some contain nearly half of training data, whereas some purely consist of test data. Hence, sample selection bias is clearly revealed in Figure 1(b), and it is obviously the culprit of the imprecise classification result shown in Figure 1(a). Furthermore, in this paper, in order to explore structures of different distributions, the number of clusters in Figure 1(b) is automatically determined by the proposed algorithm under some preference criteria (see Section 3.2 for more details).

Aiming at correcting the sample imbalance in Figure 1(b), we re-select samples from each cluster under the same proportion, e.g., 30%, as shown in Figure 1(c). It is important to notice that some clusters may consist of so few labeled data that some unlabeled ones may have to be selected into the new sample set to balance the distribution. For each of these unlabeled instances, we will mark it with "the most likely label" on the basis of its nearest neighbors. Importantly, its nearest neighbors may either be in the same cluster or different clusters. Hence, even if the samples are obtained from the clusters that do not have any labeled instance, our approach can still work well by assigning them with the most likely labels from adjacent clusters. But on the other hand, to avoid using the mislabeled samples, a carefully designed selection strategy is proposed to select those examples that "most probably have correctly assigned labels". This criterion is formally discussed and proven in Lemma 3.2. Therefore, after these procedures, the new sample set can reflect the true distribution $P(\mathbf{x}, y)$, since the new samples are uniformly distributed in the dataset shown as Figure 1(c). As constructed from this re-balanced sample set, the linear classifier can obviously construct a more accurate model as shown in Figure 1(d). As a summary, the following are the main steps of the proposed method.

- If the samples were unbiased, examples ought to be uniformly distributed in any local space of the dataset, without bias on anyone. We provide a formal proof in Lemma 3.1.

- Automatic clustering is utilized to find out the natural structure of the dataset by similarity.

- Based on the principle of uniformity, samples are evenly obtained from each cluster.

- In order to sample uniformly, some unlabeled data may be involved in the new sample set. Each unlabeled example will be given the same class label as that of its labeled nearest neighbor.

- In order to avoid using mislabeled examples, a strategy is designed to select the data which have highest probability to have the same class label as its labeled nearest neighbor's, proven in Lemma 3.2.

As described above, clustering is utilized to explore the natural structure of a dataset by similarity. But clustering is not the only way that can explore the structure of a dataset. For example, a tree can also be used to discover the hierarchical structure. But in this paper, the advantage of clustering is that it is more convenient to select the representative samples in the re-balancing step, to be discussed in Section 3.3. In addition, since the number of clusters are automatically determined by the algorithm, we also report an experimental study to explore the performance when the number of clusters are different from the choice made by the algorithm, in order to demonstrate the optimality of the criterion incorporated into the algorithm. Details can be found in Table 6. It is also important to note that our proposed method can generate uniform training set regardless of the sample size. This is where the proposed method is superior to the simple process that randomly select samples from the whole dataset, in that if the sample size is very limited, the selected chance should give more to the vital data according to the structure, whereas random selection means every data has the same opportunity every time.

3.2 Structure Discovery In our approach, clustering is chosen to discover the structure of $P(\mathbf{x}, y)$. Through clustering, a dataset X is partitioned into K clusters $C_1, C_2, \ldots, C_K$, where $\bigcup_{i=1}^{K} C_i = X$ and $C_i \bigcap C_j = \emptyset$ $(i, j = 1, 2, \ldots, K$ and $i \neq j)$. We guarantee in our specific clustering algorithms that examples inside each cluster are very similar in their distribution. Thus, intuitively, if examples are evenly sampled from each of these clusters, the data sample should not have any sample selection bias.

LEMMA 3.1. *Given a dataset X and its clusters $C_1, C_2, \ldots, C_K$, if for any $i = 1, 2, \ldots, K$, we have $P(s|C_i, y) = P(s) = \alpha$, then $P(s) = P(s|X, y)$. This implies that the dataset is unbiased.*

Proof. In Lemma 3.1,
Because $P(s|C_i, y) = P(s)$
According to the definition of conditional probability,
$P(s, C_i, y) = P(s) \times P(C_i, y)$
On the other hand,
$\bigcup_{i=1}^{K} C_i = X$
Hence, according to the definition of total probability,
$P(s, X, y) = \sum_{i=1}^{K} P(s, y, C_i)$
$= \sum_{i=1}^{K} (P(s) \times P(y, C_i))$
$= P(s) \times \sum_{i=1}^{K} P(y, C_i)$
$= P(s) \times P(y, X)$
Consequently, $P(s) = \frac{P(s, X, y)}{P(X, y)} = P(s|X, y)$

Lemma 3.1 indicates that if we can obtain samples from each cluster under the same proportion $P(s) = \alpha$, the whole sample set ought to be unbiased according to definition. In a biased dataset, the proportions of labeled data are different from cluster to cluster. Intuitively, different clusters respond differently to some biased sampling process. In practice, Lemma 3.1 provides an approach to generate unbiased sample set, i.e., sampling equal proportion from each cluster. This straightforward process does not directly work on either the biased distribution $Q(\mathbf{x}, y)$ or the unbiased $P(\mathbf{x}, y)$. In other words, our method is independent of the exact type of sample selection bias, defined between $Q(\mathbf{x}, y)$ and $P(\mathbf{x}, y)$. This is exactly where the proposed method differs from the previous approaches, most of which are trying to correct bias by learning $\frac{P(\mathbf{x},y)}{Q(\mathbf{x},y)}$.

In this paper, in order to use structure discovery to reduce bias, two traditional clustering algorithms are modified into the proposed framework and their main difference is how similarity is defined. But it is important to point out that there are other clustering algorithms that could also be utilized in our approach. But for the limited space, we don't extend them in this paper.

Modified Bisecting K-Means Bisecting K-means is a simple and effective clustering algorithm which have both the characteristic of hierarchical and partitioned. It will first split the whole dataset into two clusters, select one of these clusters to split, and so on, until K clusters have been produced [8]. But to make it effective to explore the structure of a dataset, some modification is necessary. Ideally the number of clusters, K, should not be a user specified parameter. It is impractical for the users set K precisely to explore the structure. For this purpose, two equations are introduced to control K automatically:

DEFINITION 3.1. *Given a dataset C and its two subset C_1, C_2, where $C_1 \bigcup C_2 = C$ and $C_1 \bigcap C_2 = \emptyset$, Then*

$$(3.1) \quad Par(C, C_1, C_2) = Sgn(SSE(C) - \sum_{i=1,2} SSE(C_i))$$

where $Sgn(x)$ returns 1 if x is positive and SSE is sum of squared error.

If $Par(C, C_1, C_2) = 1$, it means that we should go on dividing C into C_1 and C_2, since it can induce a smaller SSE which Bisecting K-means is seeking for. On the other hand, we bring another equation that can control K by labeled dataset L, as shown in Eq. (3.2).

DEFINITION 3.2. *Given a set C_i and labeled set L, Then*

$$(3.2) \ Purity(C_i) = \max(\frac{|C_i \bigcap L^+| + \sigma}{|C_i \bigcap L| + \sigma}, \frac{|C_i \bigcap L^-| + \sigma}{|C_i \bigcap L| + \sigma})$$

where $\sigma = 1.0 \times 10^{-3}$, in case of $|C_i \bigcap L| = 0$.

Input: Unlabeled dataset (test dataset) U; Labeled dataset (sample set) L.
Output: $C = \{C_1, \cdots, C_i, \cdots\}$: clusters
1 $C \leftarrow \emptyset$
2 $A \leftarrow U \bigcup L$
3 $C \leftarrow \{A\}$
4 **for** *each $C_i \in C$* **do**
5   Split C_i into two clusters C_{i1} and C_{i2}
6   **if** $Purity(C_i) < 0.9$ *or* $Par(C_i, C_{i1}, C_{i2}) = 1$ **then**
7   Replace C_i with C_{i1} and C_{i2}
8 **end**
9 Return clusters C

Algorithm 1: Structure Discovery via Modified Bisecting K-means (BRSD-BK).

$Purity(C_i)$ reflects the accuracy of labeled data in set C_i. For example, if C_i contains 8 positive labeled data and 2 negative labeled data, then $Purity(C_i) = max(0.8, 0.2) = 0.8$, which is also the accuracy of positive labeled data in C_i. In particular, if C_i does not have any labeled data, $Purity(C_i) = \frac{\sigma}{\sigma} = 1$. Hence, if set C_i has a low $Purity(C_i)$, it should be split further because some instances with different class labels have been in C_i. Empirically, we believe $Purity(C_i) > 0.9$ is strong enough to indicate that C_i is compact enough. But different values of $Purity(C_i)$ may lead to different number of clusters (K). In this paper, we also report an extra experiment to explore the performance of the proposed approach with different value of K in Table 6 of Section 5.

With these two equations, we modify Bisecting K-means as Algorithm 1. It starts with the whole dataset $U \cup L$ and then uses K-Means to divide the subset C_i into two clusters in step 5. If C_i triggers one of the two conditions described in step 6 (concerned with Eq. 3.1 and Eq. 3.2), it will be replaced with its two sub-clusters.

Modified DBSCAN Other than Bisecting K-means, another traditional clustering algorithm, DBSCAN, is also introduced to discover the structure of a dataset. DBSCAN is a traditional density-based clustering algorithm. Furthermore, its density is also calculated by similarity, which we can use to reveal the structure of a dataset. However, traditionally, DBSCAN causes issues on how to determine two of its user-specified parameters *Eps* and *MinPts*, a distance parameter and threshold parameter respectively. But in the proposed algorithm, these two parameters will be calculated by labeled dataset L. For *Eps*, we derive an equation that

$$(3.3) \qquad Eps = \frac{1}{3}(\mu + \hat{\mu} + \omega)$$

$$\mu = \underset{(\mathbf{l_a} \in L^+) \wedge (\mathbf{l_b} \in L^-)}{argmin} \|\mathbf{l_a}, \mathbf{l_b}\|$$

$$\hat{\mu} = \underset{(y_s = y_t) \wedge (\mathbf{l_s} \neq \mathbf{l_t})}{\operatorname{argmin}} \|\mathbf{l_s}, \mathbf{l_t}\|$$

$$\omega = \sum_{i=+,-} ((|L^i|^2 - |L^i|))^{-1} \sum_{(\mathbf{l_m^i}, \mathbf{l_n^i} \in L^i)} \|\mathbf{l_m^i}, \mathbf{l_n^i}\|)$$

In other words, μ indicates the shortest distance between different classes; $\hat{\mu}$ presents the shortest distance in the same class, while ω indicates the average distance also in the same class. All of the three parameters can be calculated by the sample set L.

Moreover, with *Eps*, we can recognize neighbors of each instance, defined by that the distances$<$ *Eps*. Then it is easy to determine *MinPts*. In the proposed algorithm, we first calculate the number of neighbors of each instance. Then

$$(3.4) \qquad MinPts = \frac{1}{2} \times A_n$$

A_n presents the average number of neighbors of all instances. With *Eps* (Eq. 3.3) and *MinPts* (Eq. 3.4), the rest of the algorithm is the same as the original DBSCAN, except that we only connect the closest neighbor to generate smaller clusters because it will be more convenient to select representative samples, which is discussed in the following section.

3.3 Re-balancing from the Structure In order to generate a uniform and balanced sample set from the structure, the samples drawn from each cluster should be under the same proportion according to Lemma 3.1. To balance the distribution, some unlabeled data may be involved in the new sample set. In our approach, we design an easy way to assign label to each of the unlabeled ones with that of its labeled nearest neighbor's. But it is really risky to use the labeled nearest neighbor to predict the unlabeled one due to the cost and the unpredictable false which may be caused by the prediction. Suppose that a pivotal sample is given the wrong class label, dozens of unlabeled ones that are akin to it may be categorized into the wrong class too. Therefore, in the proposed method, we prefer to select those instances which have higher probability to have the same label as its labeled nearest neighbor's, and then the more representative ones among them will be obtained into the new sample set, which has been illustrated as Figure 2.

This principle is also mentioned in Section 5, where a related experiment will be reported in Table 7. According to this strategy, Eq. (3.5) reflects the probability of an unlabeled instance $\mathbf{x_i}$ to have the same label as its labeled nearest neighbor's.

DEFINITION 3.3. *Given an instance* $\mathbf{x_i}$ *and labeled dataset L, then*

$$(3.5) \qquad LN(\mathbf{x_i}) = \underset{t \in \{1, \cdots, |L|\}}{\operatorname{argmax}} \frac{1}{\|\mathbf{x_i}, \mathbf{l_t}\| + \sigma},$$

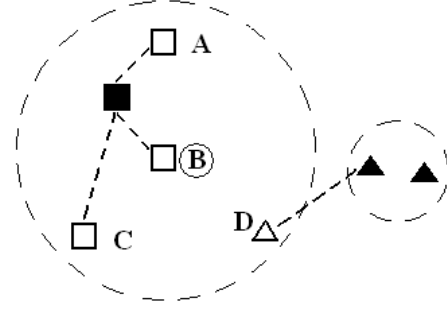*where* $\sigma = 1.0 \times 10^{-3}$, *in case of* $\|\mathbf{x_i}, \mathbf{l_t}\| = 0$.

Figure 2: Label and sample selection strategy
"▲" and "■" are two labeled samples. A cluster is depicted as a dotted circle. Unlabeled examples A, B, C are labeled as "□" by their labeled nearest neighbor "■", while D is labeled as "△" by its nearest neighbor "▲". Sample B will be selected by the algorithm because 1) it is more likely to have been assigned the correct label than both C and D, and 2) more representative than A (closer to the center of the cluster).

LEMMA 3.2. *Given an instance* $\mathbf{x_i}$ *and its nearest labeled neighbor* $\mathbf{l_n}$, *the probability of* $\mathbf{x_i}$ *has the same class label as that of* $\mathbf{l_n}$ *satisfies that*

$$(3.6) \qquad P(I|\mathbf{x_i}, \mathbf{l_n}) \propto \frac{1}{\|\mathbf{x_i}, \mathbf{l_n}\|} \propto LN(\mathbf{x_i})$$

Proof. Lemma 3.2

Assume that for a very small real number d, if $\mathbf{a}, \mathbf{b} \in \mathbf{X}$ and $\|\mathbf{a}, \mathbf{b}\| \leq d$, then $P(I|\mathbf{a}, \mathbf{b}) = \beta \to 1$.

We can generate a sequence $\{\mathbf{m_0}, \mathbf{m_1}, \cdots, \mathbf{m_k}\}$, where $\mathbf{m_0} = \mathbf{x_i}$ and $\mathbf{m_k} = \mathbf{l_n}$ and for any integer $0 \leq t < k$, $\|\mathbf{m_t}, \mathbf{m_{t+1}}\| = d$. Then we can get $k_{min} = \frac{\|\mathbf{x_i}, \mathbf{l_n}\|}{d}$.

Moreover, because $P(I|\mathbf{m_t}, \mathbf{m_{t+1}}) = \beta \to 1$, we can use $\mathbf{m_t}$ to predict $\mathbf{m_{t-1}}$, then the probability of $\mathbf{m_0}$ has the same label as $\mathbf{m_k}$ can be described as:

$$P(I|\mathbf{m_0}, \mathbf{m_k}) \leq \prod_{t=1}^{k_{min}} P(I|\mathbf{m_t}, \mathbf{m_{t-1}}) = \beta^{\frac{\|\mathbf{x_i}, \mathbf{l_n}\|}{d}}$$

On the other hand, $0 < \beta < 1$,

Therefore, $P(I|\mathbf{x_i}, \mathbf{l_n})_{max} = \beta^{\frac{\|\mathbf{x_i}, \mathbf{l_n}\|}{d}} \propto \frac{1}{\|\mathbf{x_i}, \mathbf{l_n}\|} = \underset{t}{\operatorname{argmax}} \frac{1}{\|\mathbf{x_i}, \mathbf{l_t}\|} \propto LN(\mathbf{x_i})$

Lemma 3.2 indicates that the probability of an instance $\mathbf{x_i}$ to have the same class label as its labeled nearest neighbor's can be reflected by $LN(\mathbf{x_i})$. Hence, we should select those $\mathbf{x_i}$ which have higher value of $LN(\mathbf{x_i})$. It is important to note that any labeled instances $\mathbf{x_i}$ have a large value of $LN(\mathbf{x_i}) = \frac{1}{\sigma}$. This is reasonable because they already have the correct class labels. On the other hand, another equation can be defined to indicate the "representative" of an instance $\mathbf{x_r}$.

```
Input: The $i$th cluster $C_i$; Labeled dataset $L$;
       Expected proportion of samples $\alpha$.
Output: $S_i$: sample set drawn from $C_i$
1  $n \leftarrow \alpha \times |C_i|$
2  $S_i \leftarrow \emptyset$
3  $N = \max(n, |C_i \cap L|)$
4  if $|C_i| = 1$ then
5    Return $S_i$
6  else
7    Sort all the $\mathbf{x_t} \in C_i$ by $LN(\mathbf{x_t})$ (Eq. 3.5), in
     descending order.
8    $P_i \leftarrow$ The first $N$ sorted instances $\{\mathbf{x_1}, \cdots, \mathbf{x_N}\}$
9    Sort all the $\mathbf{x_r} \in P_i$ by $RP(\mathbf{x_r})$ (Eq. 3.7), in
     descending order.
10   $S_i \leftarrow$ The first $n$ sorted instances $\{\mathbf{x_1}, \cdots, \mathbf{x_n}\}$
11   Label the unlabeled instances $\mathbf{x_u} \in S_i$ with its
     nearest labeled neighbor
12   Return $S_i$
```

Algorithm 2: Re-balancing by sample selection from each cluster.

DEFINITION 3.4. *Given a cluster C_i and $\mathbf{x_r} \in C_i$, then*

$$(3.7) \qquad RP(\mathbf{x_r}) = \frac{1}{\|\mathbf{x_r}, \mathbf{c_i}\|}$$

Eq. (3.7) indicates that in a cluster, the most representative instance should be the one close to the center of the cluster. This strategy is also often utilized in prototype selection [10]. With Eq. (3.5) and Eq. (3.7), the re-balancing strategy can be described as Algorithm 2. In Algorithm 2, we will not draw samples from the one-instance cluster to avoid noisy data in the new sample set (in step 4 and step 5). For a multi-instance cluster, an instance set P_i is first generated by the more predictable instances (higher value of $LN(\mathbf{x_r})$) in Step 6. And then the more representative ones among them are selected into the new sample set. In Algorithm 2, the labeled data seem to have more chance to be selected into the corrected sample set because they have a high value of $LN(x_r)$. But actually, most of them will not be chosen by the strategy, because the number of samples drawn from each cluster is limited by α while some clusters contain too much labeled data, which is also discussed in Figure 1 of Section 3.1 and Lemma 3.1 of Section 3.2. In addition, an experimental study is reported on this behavior in Table 7 of Section 5, in order to study how sample selection bias is reduced by avoiding redundant labeled instances into the corrected sample set.

4 Experiments

We have chosen a mixture of problems from UCI, varying in number of classes, number of examples, types and number of features, summarized in Table 2. In order to isolate the effect of sampling size, the training set size of the same problem under different bias is chosen to be the same. In order to test the "goodness" of corrected samples across different learning algorithms, we have chosen four different inductive learners: C4.5, naive Bayes, NNge or nearest neighbor, and logistic regression. For short, we call the proposed algorithm BRSD or Bias Reduction via Structure Discovery, and the implementations using modified Bisecting K-means and modified DBSCAN as BRSD-BK and BRSD-DB.

4.1 Feature Bias Reduction We directly follow the definition of "feature bias" $P(s = 1|\mathbf{x}, y) = P(s = 1|\mathbf{x})$, to generate feature bias from every dataset. To do so, we first randomly select 50% of the features, and then we sort the dataset according to each of the selected features (dictionary sort for categorical and numerical sort for continuous). Finally, we attain top instances from every sorted list, with the total sample size of "#Sample" as described in Table 2.

The experiment results can be found in Table 3. The numbers under the column "Bias" are the accuracies of classifiers on the original biased dataset, and the numbers under "BK" and "DB" are the results of the classifiers that use corrected samples generated by either modified Bisecting K-means or modified DBSCAN. Additionally, the improvement in accuracy using C4.5 as based learner is plotted in Figure 3.

Both BRSD-BK and BRSD-DB have performed higher accuracies (10% to 50% higher in most cases) for each inductive learners (C4.5, naive Bayes, etc) than the corresponding classifiers constructed on the biased datasets, implying our proposed method can evidently overcome feature bias. For example, the accuracies of multi-class dataset "Letter", which categorizes handwriting patterns from different people into 26 classes (letters), has been improved from 53% to 80% when it is performed on C4.5 after the use of BRSD-DB. For the two clustering algorithm in the proposed method, the accuracies of corrected samples generated by BRSD-DB appears higher than BRSD-BK.

4.2 Class Bias Reduction For class bias, or $P(s = 1|\mathbf{x}, y) = P(s = 1|y)$, which indicates the situation that the training data have different prior class probability distribution from the true probability distribution, we simulate it by the following steps: (1) randomly generate prior class probability for the dataset, with one dominating class under the prior probability between 60% and 80%; (2) attain samples from every classes under the probability generated in (1), with a total sample size of "#Sample" described in Table 2. The results can be found in Table 4, and the accuracy improvement of naive Bayes is plotted in Figure 4.

Comparing Table 4 with Table 3, the effect of class bias on model's accuracy is more significant than that of feature bias. For the "Letter" dataset, C4.5's accuracy drops from

Table 2: Description of the datasets

Order	Dataset	#Instance	#Feature	#Class	#Sample	# Unlabeled
1	SatImage	6435	37	6	180	6255
2	Segment	2310	19	7	90	2220
3	Ionosphere	351	34	2	34	317
4	Iris	150	4	3	20	130
5	Letter	20000	17	26	300	19700
6	Optdigits	5620	65	10	60	5560
7	Wine	178	13	3	15	163
8	ColonTumor	62	2000	2	15	47
9	Diabetes	768	8	2	80	688
10	Glass	214	9	7	20	194
11	Haberman	306	3	2	30	276
12	Mfeat	2000	649	10	60	1940
13	Wdbc	569	30	2	30	539
14	Sonar	208	60	2	20	188
15	Spambase	4601	57	2	25	4580
16	Vehicle	846	18	4	84	762
17	WavForm	5000	40	3	75	4925

Note: Three kinds of biased training set as well as the new sample set will have the same sample size (#Sample) described in Table 2.

Table 3: Accuracy of feature biased training set and BRSD (BK and DB)

Datasets	C4.5			Naive Bayes			NNge			Logistic Regression		
	Bias	BK	DB	Bias	BK	DB	Bias	BK	DB	Bias	BK	DB
SatImage	0.25	0.67	**0.75**	0.40	0.78	**0.79**	0.44	0.77	**0.83**	0.39	0.73	**0.83**
Segment	0.45	0.83	**0.99**	0.49	0.72	**0.97**	0.56	0.86	**1.00**	0.64	0.81	**1.00**
Ionosphere	0.36	0.74	**0.89**	0.39	0.76	**0.85**	0.34	0.81	**0.90**	0.37	0.78	**0.86**
Iris	0.57	0.83	**0.93**	0.64	0.89	**0.94**	0.89	0.92	**0.96**	0.73	0.91	**0.93**
Letter	0.53	0.66	**0.80**	0.55	0.75	**0.81**	0.63	0.74	**0.84**	0.67	0.75	**0.89**
Optdigits	0.56	0.56	**0.73**	0.62	0.71	**0.84**	0.70	0.79	**0.84**	0.77	0.83	**0.90**
Wine	0.50	**0.90**	0.80	0.63	0.74	**0.88**	0.62	**0.87**	0.87	0.59	0.68	**0.91**
ColonTumor	0.45	**0.66**	0.62	0.70	**0.77**	0.77	0.66	**0.72**	0.68	0.66	**0.72**	0.66
Diabetes	0.66	**0.73**	0.72	0.64	**0.72**	0.72	0.65	0.73	**0.74**	0.63	**0.73**	0.73
Glass	0.35	**0.49**	0.43	0.44	**0.52**	0.48	0.54	**0.67**	0.51	0.39	0.43	**0.47**
Haberman	0.73	0.73	0.73	0.50	**0.74**	0.71	0.52	0.72	**0.74**	0.55	0.74	**0.75**
Mfeat	0.64	0.63	**0.72**	0.47	0.67	**0.74**	0.57	0.73	**0.78**	0.82	0.81	**0.83**
Wdbc	0.61	0.87	**0.88**	0.73	0.87	**0.91**	**0.89**	0.86	0.88	0.65	**0.91**	0.81
Sonar	0.59	**0.69**	0.61	0.67	0.75	**0.77**	0.71	**0.79**	0.74	0.64	**0.71**	0.70
Spambase	0.72	**0.76**	0.76	0.73	**0.83**	0.83	0.71	**0.76**	0.76	0.70	**0.80**	0.80
Vehicle	0.46	0.46	**0.57**	0.41	0.44	**0.46**	0.43	0.56	**0.67**	0.58	0.62	**0.67**
WaveForm	0.67	0.67	**0.70**	0.79	0.79	**0.83**	0.72	0.75	**0.77**	0.75	0.78	**0.84**

Note: For each classifier, the highest accuracy of the according dataset is highlighted in bold.

53.34% on feature bias down to 7.44% on class bias. This is obviously due to the effect of class bias that significantly modifies the right number of examples with different class labels. Importantly however, after the use of the proposed algorithms, the classifiers can obviously obtain higher accuracies, especially for the multi-class problems, such as "Letter", "Mfeat" and "Optdigits", with an improvement ranging from 30% to 60%.

4.3 Complete Bias Reduction Complete bias chooses examples according to both the feature vectors and class labels. Since it is difficult to justify the utility of any single mapping function to generate complete bias across different datasets, we use the bootstrap sampling method as previously adopted in [4] to simulate complete bias. Thus, some labeled instances $(\mathbf{x}, y)$ are more frequent than others and vice versa in the biased sample, which carries a different distribution from the unbiased distribution.

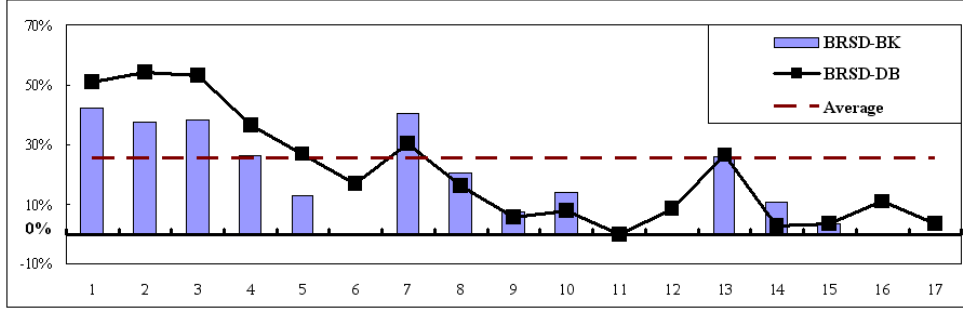The experiment results are summarized in Table 5.

Figure 3: Accuracy Increase ($ACC_{BRSD} - ACC_{Bias}$) of C4.5 using BRSD-BK and BRSD-DB with Feature Bias, and average improvement of both BRSD-BK and BRSD-DB for 17 datasets. The x-axis is the id of each dataset.

Table 4: Accuracy of class biased training set and BRSD (BK and DB)

Dataset	C4.5			Naive Bayes			NNge			Logistic Regression		
	Bias	BK	DB	Bias	BK	DB	Bias	BK	DB	Bias	BK	DB
SatImage	0.43	0.68	**0.70**	0.44	**0.75**	0.62	0.44	**0.71**	0.68	0.51	**0.75**	0.69
Segment	0.28	0.79	**0.82**	0.28	0.69	**0.73**	0.54	0.74	**0.87**	0.33	0.86	**0.92**
Ionosphere	0.40	0.63	**0.70**	0.39	0.68	**0.74**	0.36	**0.71**	0.66	0.55	0.62	**0.75**
Iris	0.78	**0.94**	0.88	0.66	**0.95**	0.95	0.75	0.97	**0.98**	0.74	0.95	**0.98**
Letter	0.07	0.26	**0.39**	0.08	0.29	**0.48**	0.12	0.28	**0.49**	0.24	0.38	**0.56**
Optdigits	0.23	0.36	**0.75**	0.19	0.49	**0.84**	0.18	0.52	**0.86**	0.19	0.58	**0.90**
Wine	0.30	0.54	**0.87**	0.23	0.56	**0.92**	0.48	0.57	**0.85**	0.58	0.57	**0.85**
ColonTumor	0.68	0.68	0.68	0.72	**0.81**	0.79	0.79	0.64	**0.85**	**0.83**	0.70	0.72
Diabetes	0.74	0.72	**0.76**	0.69	**0.73**	0.72	0.68	0.73	**0.74**	0.69	0.71	**0.74**
Glass	0.48	**0.56**	0.32	0.19	0.50	**0.51**	0.24	**0.48**	0.41	0.24	**0.55**	0.55
Haberman	0.26	0.50	**0.68**	0.27	**0.75**	0.74	0.26	0.53	**0.72**	0.27	0.59	**0.76**
Mfeat	0.32	0.46	**0.61**	0.08	0.24	**0.67**	0.17	0.37	**0.69**	0.41	0.51	**0.79**
Wdbc	**0.90**	0.90	0.88	0.93	0.92	**0.94**	0.93	0.94	**0.95**	0.87	0.90	**0.91**
Sonar	0.44	**0.58**	0.57	0.49	**0.61**	0.61	0.49	**0.59**	0.57	0.49	**0.58**	0.48
Spambase	0.77	**0.85**	0.85	**0.81**	0.76	0.80	0.61	0.75	**0.79**	0.82	**0.86**	0.80
Vehicle	0.40	0.49	**0.50**	0.35	0.43	**0.51**	0.30	0.46	**0.49**	0.48	0.54	**0.55**
WaveForm	0.55	0.62	**0.68**	0.59	0.65	**0.81**	0.52	0.60	**0.67**	0.57	0.69	**0.83**

Note: For each classifier, the highest accuracy of the according dataset is highlighted in bold.
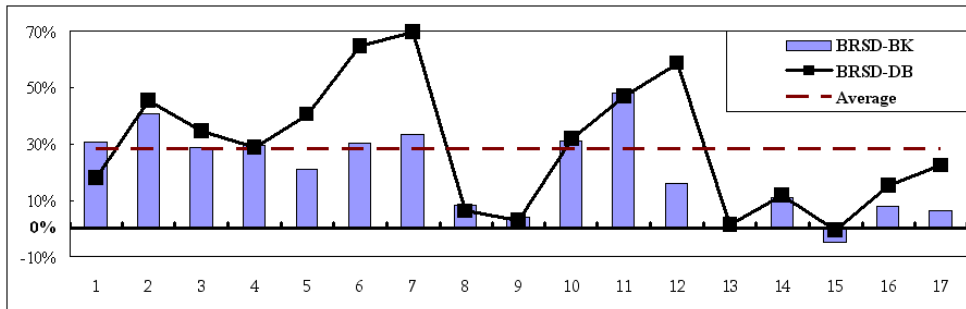

Figure 4: Accuracy Increase ($ACC_{BRSD} - ACC_{Bias}$) of Naive Bayes using BRSD-BK and BRSD-DB with Class Bias, and average increase of both BRSD-BK and BRSD-DB for 17 datasets. The x-axis is the id of each dataset.

Moreover, the accuracy comparison between the original biased dataset and the corrected sample set is plotted in Figure 5 on nearest neighbor and logistic regression, where x-axis is the accuracy on biased sample and y-axis is the accuracy on corrected sample. Similar to earlier results on other types of bias, the accuracies of classifiers have been

Table 5: Accuracy of complete biased training set and BRSD (BK and DB)

Dataset	C4.5			Naive Bayes			NNge			Logistic Regression		
	Bias	BK	DB	Bias	BK	DB	Bias	BK	DB	Bias	BK	DB
SatImage	0.44	**0.72**	0.67	0.33	**0.80**	0.71	0.42	**0.74**	0.68	0.45	**0.78**	0.72
Segment	0.25	0.79	**0.89**	0.28	0.70	**0.74**	0.54	0.77	**0.85**	0.29	0.80	**0.87**
Ionosphere	0.45	0.63	**0.70**	0.37	0.68	**0.72**	0.31	**0.71**	0.56	0.62	0.62	**0.78**
Iris	0.61	0.77	**0.83**	0.61	0.84	**0.95**	0.62	**0.96**	0.95	0.62	0.95	**0.96**
Letter	0.08	0.30	**0.50**	0.06	0.29	**0.54**	0.09	0.30	**0.55**	0.12	0.38	**0.61**
Optdigits	0.25	0.42	**0.75**	0.16	0.46	**0.86**	0.12	0.40	**0.78**	0.33	0.59	**0.91**
Wine	0.30	0.54	**0.81**	0.22	0.62	**0.95**	0.53	0.74	**0.90**	0.38	0.67	**0.83**
ColonTumor	0.40	**0.68**	0.57	0.57	**0.62**	0.57	**0.64**	0.55	0.62	0.55	**0.68**	0.57
Diabetes	0.66	**0.72**	0.71	0.70	**0.73**	0.69	0.68	**0.75**	0.74	0.70	**0.73**	0.73
Glass	0.47	0.47	**0.58**	0.10	0.39	**0.45**	0.08	**0.46**	0.41	0.14	0.41	**0.47**
Haberman	0.20	0.74	**0.77**	0.20	0.72	**0.77**	0.20	0.38	**0.64**	0.20	0.59	**0.77**
Mfeat	0.17	0.60	**0.68**	0.08	0.30	**0.69**	0.12	0.36	**0.70**	0.33	0.75	**0.87**
Wdbc	**0.90**	0.89	0.82	**0.91**	0.90	0.91	0.91	**0.93**	0.91	0.87	**0.89**	0.88
Sonar	0.46	0.49	**0.60**	0.50	0.51	**0.69**	0.50	0.51	**0.54**	0.46	0.53	**0.65**
Spambase	0.78	0.78	**0.80**	**0.85**	0.85	0.84	0.78	**0.81**	0.72	0.80	0.81	**0.85**
Vehicle	0.31	0.42	**0.48**	0.29	0.35	**0.51**	0.30	0.40	**0.44**	0.42	0.46	**0.63**
WaveForm	0.55	0.53	**0.69**	0.59	0.60	**0.80**	0.50	0.55	**0.76**	0.59	0.55	**0.85**

Note: For each classifier, the highest accuracy of the according dataset is highlighted in bold.
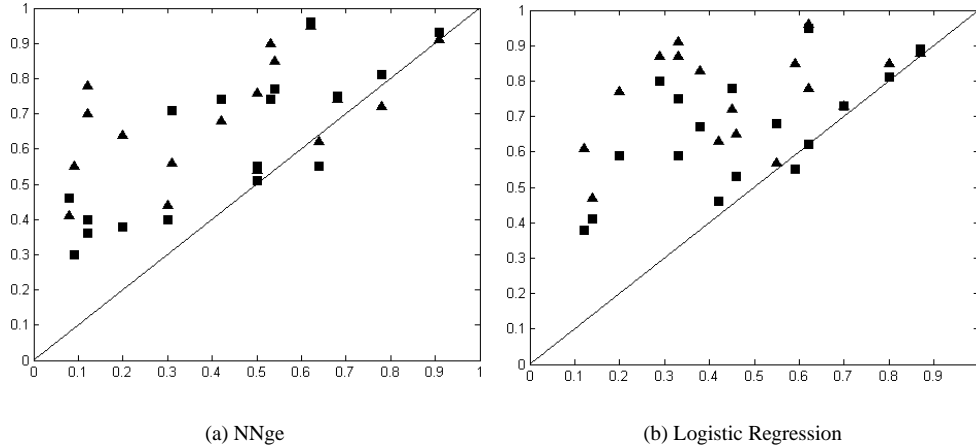

(a) NNge

(b) Logistic Regression

Figure 5: Accuracy comparison between complete biased training set and BRSD, on NNge and Logistic Regression. The x-axis is the accuracy of the according classifier on the complete biased training set; the y-axis is the accuracy of the according classifier on the corrected sample set generated by BRSD. In particular, BRSD-BK is plotted as "■", while BRSD-DB is plotted as "▲". All the dots above "$y = x$" indicates the corrected sample set can get a higher accuracy.

increased by 10% to 60% on the corrected sample set generated by BRSD-BK and BRSD-DB. The scatter plots visually show the number of wins/losses across all comparisons. For nearest neighbor, across 34 pairwise comparisons, 31 out of 34, the corrected sample improves accuracy, and for logistic regression, the improvement happens in 33 out of 34 times.

5 Discussions

We mainly discuss on the following questions:

- How sensitive is the number of clusters K? Is the chosen number by the algorithm reasonably optimal?

- What is the composition of the corrected sample set? How does the proposed method reduce sample selection bias by avoiding too much labeled and biased instances into the corrected sample set?

- How does the proposed method perform with the increasing size of the corrected sample set?

Table 6: Performance under different number of clusters (K)

K	#Sample	#L (Total: 90)	ACC (C4.5)
1	90	90	0.28
18	90	46	0.43
36	90	36	0.47
91	90	34	0.56
313	90	34	0.64
826*	**90**	**33**	**0.79**
1700	90	37	0.56

Note: Table 6 reports the performance of BRSD-BK with different number of clusters. The experiment is performed on "Segment" with class bias, which has 2310 instances and 90 samples. "#L" indicates the number of original labeled data that have been included in the new sample set. The number of clusters explored by Algorithm 1 is highlighted in bold (*).

Table 7: Performance under different sample size (#Sample)

#Sample	#L (Total: 90)	#Un	#Wrong	ACC (C4.5)
49	25	24	3	0.64
66	29	37	5	0.74
90	33	57	3	0.79
128	43	85	10	0.62
201	56	145	12	0.82
438	64	374	92	0.69

Note: #L indicates the number of original labeled data in the new sample set; #Un indicates the number of unlabeled data in the new sample set; #Wrong indicates the number of unlabeled example with mistakenly predicted class labels.

In the proposed approach, clustering is utilized to explore the structure of a dataset by similarity. Moreover, the number of clusters is determined by the algorithm under some preference criteria. We setup an additional experiment using BRSD-BK, by modifying Algorithm 1 to control the number of clusters. For example, we can change step 6 to $Purity(C_i) < 0.5$ to obtain fewer clusters. This can be achieved since more labeled data with different labels are now allowed to be in one cluster. The performance as a function of increasing number of clusters is reported in Table 6.

Table 6 reports the experiment results on "Segment" dataset under class bias with the increasing number of clusters (K). But the sample set (#Sample) remains the size of 90 in order to isolate the effect of sampling size. In Table 6, besides the accuracies of C4.5, the number of original labeled data in the corrected sample set is also reported under "#L". Through this table, We can observe that with the increasing value of "K", the number of original labeled data in the new sample set (#L) decreases in the beginning, and then reaches a relatively stable value around 34. But the result of C4.5 is more and more accurate before K reaches 1700. We own these increase in accuracy to the advantage of small clusters, which can explore more subtle structures in a dataset. In other words, by generating small clusters, only the "most similar" instances will be grouped in one cluster. This is useful in the re-balancing step (Algorithm 2), because the samples selected from these compact groups may be more representative. But by observing the last row, when we generate 1700 clusters, it is no longer the case that we can improve the accuracy. Instead, the number of original labeled data increases to 37 and the accuracy drops to 0.56. This is because that when we only have "tiny" clusters, the effect of Eq. (3.5) dominates that of Eq. (3.7). This means that the labeled and biased instances have more chance to be included into the new sample set (Algorithm 2).

On the other hand, as discussed in Section 3.3, the labeled data seem to have more chance to be selected into the new sample set by Algorithm 2, because they have higher values of $LN(x_r)$. But through Table 7, which summarizes the performance on the "Segment" dataset as sample size increases, we can observe that only 20%~60% of the labeled data (#L) are included into the new sample set even if sample size (#Sample) is large (nearly 5 times of the original size). On the contrary, 50%~85% of the corrected samples are unlabeled. This is because the number of samples drawn from each cluster is limited by α, while some clusters contain so many labeled and biased instances that most of these biased samples will not be chosen into the new sample set. This is exactly one of the important reasons that sample selection bias can be effectively reduced in our algorithm, which is also discussed in Figure 1 of Section 3.1 and Lemma 3.1 of Section 3.2. On the other hand, through Table 7, we observe that the effect caused by false prediction is very significant. For example, when the sample size reaches 128, 10 out of 85 unlabeled examples are given the wrong class labels, and the accuracy drops to 0.62. Hence, to avoid using too many these mislabeled examples, we have proposed Eq. (3.5) in Algorithm 2. Actually, in Table 7, although the sample size reaches 201, nearly twice as much as the former size(128), only 2 more mistakenly predicted examples are added comparing with the former ones, and the accuracy gains an improvement to 0.82. This is obviously the contribution of Eq. (3.5) and Lemma 3.2 to avoid using too much of the mislabeled samples. In addition, the contribution of Eq. (3.5) can also be reflected by the first three rows in Table 7, where "#Wrong" is stable at around 4 even sample size is increasing. But with respect to the risk of false prediction, we propose the size of the corrected sample set should be roughly the same as the original sample size in practice.

6 Related Works

The problem of sample selection bias has received much attention in machine learning and data mining. In [14] and

later [5], they introduce categorizations to present the behavior of learning algorithms under sample selection bias. In [14], an approach to correct "feature bias" by re-sampling is proposed. But 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. Later, the approach in [6] is proposed as an improvement over [14], in which, $P(s = 1|\mathbf{x})$ is estimated from both labeled and unlabeled data under the assumption that "conditional probability is not changed by feature bias". In [4], they discussed various effects of sample selection bias on inductive modeling and presented an approach based on model averaging and unlabeled data. Latter, Bickel and Scheffer [2] present a Dirichlet-enhanced process prior on top of several learning problems with related sample selection bias. Then in [1], a discriminative learning method is proposed to solve different training and test distribution. They use a kernel logistic regression approach to resolve by shifting the problem as an integrated optimization problem. In essence, much of these previous work reduce sample selection bias by learning the proportion of $\frac{P(\mathbf{x},y)}{Q(\mathbf{x},y)}$, varying from the models or the assumptions that are made. The difference of this paper with the previous work is that we discover the natural structure of the target distribution directly, by which all three types of sample selection bias can be effectively reduced. In other words, the proposed method is aimed to generate a uniformly distributed sample set, independent of any specific type of sample bias.

7   Conclusions

We propose a novel approach based on clustering to explore the intrinsic structure hidden inside of a dataset. Then, by sampling equal proportion of examples from each cluster, different types of sample selection biases can be evidently reduced. We discuss the strict criteria on how to use clustering to discover "similar" examples and group them into clusters. Specifically, these criteria are incorporated into two traditional clustering algorithms, Bisecting K-means and DB-SCAN. When unlabeled data need to be chosen to correct sample selection bias, a carefully designed sample selection strategy, orthogonal from clustering criteria, is proposed in order to choose only those examples whose "assigned labels" are most likely to be correct. There are two important advantages of the proposed algorithm. First, it is independent of any specific type of sample selection biases. This is where our approach differs from most of the previous works. Secondly, it is not built on top of any given inductive learning method but works directly on the dataset, thus, the corrected data can be used by a wide variety of inductive learning algorithms.

We simulate all three types of sample selection biases on 17 different classification problems in the experiment and choose four inductive learning algorithms. The results indicate that our proposed algorithm can significantly increase the accuracies of different classifiers (around 30% on average). Additional experiments are presented and discussed in Section 6, in order to demonstrate the advantage of those preference criteria adopted into the algorithm.

References

[1] S. Bickel, M. Brückner, T. Scheffer, *Discriminative Learning for Differing Training and Test Distributions*. In Proceedings of the 24 th International Conference on Machine Learning, ICML 2007.

[2] S. Bickel, T. Scheffer, *Dirichlet-enhanced spam filter based on biased samples*. In advances in Neural Information Processing Systems 19, pages 161-168, MIT Press, 2007.

[3] M. Dudik, R. Schapire and S. Phillips, *Correcting sample selection bias in maximum entropy density estimation*. In Advances of Neural Information Processing Systems Conference, MIT Press, 2005.

[4] W. Fan, and I. Davidson, *On Sample Selection Bias and Its Efficient Correction via Model Averaging and Unlabeled Examples*. In Proceedings of the SIAM International Conference on Data Mining, SDM 2007.

[5] W. Fan, I. Davidson, B. Zadronzny and P. Yu, *An Improved Categorization of Classifier's Sensitivity on Sample Selection Bias*. The 5th IEEE International Conference on Data Mining, ICDM 2005.

[6] J. Huang, A. Smola, A. Gretton, K. Borgwardt, and B. Scholkopf, *Correcting Sample Selection Bias by Unlabeled Data*. In Advances of Neural Information Processing System Conference, MIT Press, 2007.

[7] Heckman, J.,*Sample selection bias as a specification error*. Econometrica, 47: 153-161

[8] S. M. Savaresi and D. L. Boley, *On the performance of bisecting K-means and PDDP*. In Proceedings of the SIAM International Conference on Data Mining, SDM 2001.

[9] Chawla N. V. and Karakoulas G., *Learning From Labeled And Unlabeled Data: An Empirical Study Across Techniques And Domains*. Journal of Artificial Intelligence and Research, Volume 23, pages 331-366, 2005.

[10] Elżbieta PeRkalska, Robert P.W. Duin, Pavel Paclík, *Prototype selection for dissimilarity-based classifiers*. Journal of Pattern Recognition, Volume39, pages 189-208, 2005.

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

[12] A. Smith, C. Elkan, *Making Generative Classifiers Robust to Selection Bias*. The 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2007.

[13] A. Smith, C. Elkan, *A Bayesian network framework for reject inference*. In Proceedings of Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD 2004.

[14] B. Zadrozny,*Learning and evaluating classifiers under sample selection bias*. The 21th International Conference on Machine Learning, ICML 2004.