An Improved Categorization of Classifier's Sensitivity on Sample Selection Bias

Wei Fan[1] Ian Davidson[2] Bianca Zadrozny[1] Philip S. Yu[1]

[1] IBM T.J.Watson Research, Hawthorne, NY 10532 {weifan,psyu,zadrozny}@us.ibm.com

[2]Computer Science, State University of New York, Albany, NY 12222 davidson@cs.albany.edu

Abstract

A recent paper categorizes classifier learning algorithms according to their sensitivity to a common type of sample selection bias where the chance of an example being selected into the training sample depends on its feature vector $\mathbf{x}$ *but not (directly) on its class label* y*. A classifier learner is categorized as "local" if it is insensitive to this type of sample selection bias, otherwise, it is considered "global". In that paper, the true model is not clearly distinguished from the model that the algorithm outputs. In their discussion of Bayesian classifiers, logistic regression and hard-margin SVMs, the true model (or the model that generates the true class label for every example) is implicitly assumed to be contained in the model space of the learner, and the true class probabilities and model estimated class probabilities are assumed to asymptotically converge as the training data set size increases. However, in the discussion of naive Bayes, decision trees and soft-margin SVMs, the model space is assumed not to contain the true model, and these three algorithms are instead argued to be "global learners". We argue that most classifier learners may or may not be affected by sample selection bias; this depends on the dataset as well as the heuristics or inductive bias implied by the learning algorithm and their appropriateness to the particular dataset.*

1 Introduction

A common assumption made in data mining is that the training and test sets are drawn from the same distribution. However, in practice this rarely happens [2]. Assume that the event $s = 1$ denotes that a labeled example $(\mathbf{x}, y)$ is selected from the domain $\mathcal{D}$ of examples into the training set D, and that $s = 0$ denotes that $(\mathbf{x}, y)$ is not chosen. When constructing a classification model, we only have access to examples where $s = 1$. In [2], four different types of sample selection bias are clearly discussed according to the dependency of s on $\mathbf{x}$ and y. In this paper, we are only interested in the second case where the selection bias s is dependent on the feature vector $\mathbf{x}$ and it is conditionally independent of the true class label y given $\mathbf{x}$, i.e., $P(s|\mathbf{x}, y) = P(s|\mathbf{x})$. This type of sample selection naturally exists. For example, in direct marketing, the customers are selected into the sample based on whether or not they have received the offer in the past. Because the decision to send an offer is based on the known characteristics of the customers (that is, $\mathbf{x}$) before seeing the response (that is, y) then the bias will be of this type. In this paper, we shall focus on and refer to the second type of sample bias as it is believed to be a prevalent problem [2]. In [2], inductive learners are categorized into two types, either "local" or "global", according to their dependency/sensitivity to sample selection bias.

Definition 1.1 *The output of a* ***Local Learner*** *depends asymptotically only on* $P(y|\mathbf{x})$*. [2]*

Definition 1.2 *The output of a* ***Global Learner*** *depends asymptotically both on* $P(\mathbf{x})$ *and* $P(y|\mathbf{x})$*. [2]*

In the above definition, "the output of the learner" refers to the classifier constructed from the training set by a particular learner. These definitions were discussed in the context of the second type of sample bias [2]. In [2], several popular learners including Bayesian classifiers, naive Bayes, decision trees, logistic regressions as well as soft and hard margin SVMs, are categorized as always being either "local" or "global" . This original categorization is independent from the particular application problem and only dependent on the learner. However, several strong assumptions are implicitly made, as discussed below, regarding the model space of the classifiers and the interpretation of $P(y|\mathbf{x})$. In the discussion of Bayesian classifiers, logistic regression, and hard-margin SVM always to be sample bias independent "local" classifiers, the following two assumptions are made implicitly.

Assumption 1.1 *The learner outputs a classifier* θ *that produces the probability* $P(y|\mathbf{x}, \theta)$ *to approximate/estimate the model independent true probability* $P(y|\mathbf{x})$*.*

Assumption 1.2 *For* ***Bayesian classifiers, logistic regression, and hard-margin SVM:*** *The learner is a consistent estimator of the true probabilities* $P(y|\boldsymbol{x})$*. That is, the estimated probability* $P(y|\mathbf{x}, \theta)$ *equals the true model independent probability* $P(y|\mathbf{x})$ *when the training data is exhaustive. Formally,* $\forall \mathbf{x} \lim_{|D|\to\infty} P(y|\mathbf{x}, \theta) = P(y|\mathbf{x})$*.*

In this notation, θ denotes the classifier or "the output of the learner" constructed from training data D. (A summary of all notations is in Figure 1.) The above assumptions are strong since the individuality of different problems and different datasets is not considered. In this paper, we relax these assumptions and take these differences into account. We argue that Bayesian classifiers, logistic regression and hard-margin SVMs can be either "local" or "global" classifiers. This is dependent on the particular dataset to which these learners are applied. On the other hand, when naive Bayes, decision tree and soft-margin SVMs are characterized as "global" classifiers in [2], the following assumption is made that replaces Assumption 1.2.

Assumption 1.3 *For* ***naive Bayes, decision tree, and soft-margin SVMs:*** *The learner's model space does not contain the true model and hence is inconsistent. Formally,* $\exists \mathbf{x} \lim_{|D|\to\infty} P(y|\mathbf{x}, \theta) \neq P(y|\mathbf{x})$*.*

Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)
1550-4786/05 $20.00 © 2005 IEEE

- $\mathbf{x}$ is feature vector, y is class label, and $s = 1$ denotes that an example $(\mathbf{x}, y)$ is selected into the training set D.
- $P(s = 1|\mathbf{x}, y)$ formally describes sample selection bias, and it denotes the probability that an example $(\mathbf{x}, y)$ is selected into the training set.
- $P(s = 1|\mathbf{x}, y) = P(s = 1|\mathbf{x})$ is true for the second type of sample selection bias where s is only dependent on the feature vector $\mathbf{x}$ and it is independent from class label y.
- $P(\mathbf{x})$ is the probability distribution of feature vector $\mathbf{x}$ and it is not related to either class label y or sample selection bias s.
- $P(y|\mathbf{x})$ denotes the true conditional probability for a feature vector $\mathbf{x}$ to be a member of class y. It is completely determined by the true concept or true function that an inductive learner is to model. The true function is typically unknown unless the dataset is synthesized. $P(y|\mathbf{x})$ is independent from training data as well as sample selection bias.
- Θ is the model space assumed by a learner.
- θ is a classifier constructed by a learner by searching in the model space Θ given training data D. By definition, θ is dependent on both Θ and D.
- $P(y|\mathbf{x}, \theta)$ denotes the probability for an example $\mathbf{x}$ to be of class y, as estimated by a classifier θ. Typically, $P(y|\mathbf{x}, \theta) \neq P(y|\mathbf{x})$, in other words, the estimated probability may not be equal to the true probability.
- Since θ is dependent on both Θ and D, we define $P(y|\mathbf{x}, D, \Theta)$ to represent the same probability as $P(y|\mathbf{x}, \theta)$, i.e., $P(y|\mathbf{x}, D, \Theta) = P(y|\mathbf{x}, \theta)$.

Figure 1. Summary of Symbols and Concepts

Similarly, this assumption is also strong because the differences between datasets are not considered. We also argue that naive Bayes, decision trees and soft-margin SVMs could be either "local" (invariant to sample bias) or "global" (affected by sample bias) when the differences between datasets are accounted for. We generalize this argument that most of the known inductive learners could behave either as "local" or "global" learners, depending on the particular dataset as well as the inductive bias implied by the algorithm. When the true model of the particular dataset is contained in the model space, the learner would be local. Nonetheless, when the true model is not contained in the model space, the learner would be global.

2 Improved Categorization

Formal Definitions: In [2], $P(\mathbf{x})$ is defined as "a global distribution over the entire input space", and a global classifier is affected by sample selection bias "because the bias changes $P(\mathbf{x})$". Formally, $P(\mathbf{x})$ is the probability distribution solely as a function of the feature vector $\mathbf{x}$, and it is independent of class labels y. For example, assume that there are only two binary-valued features and each unique combination of feature values happens with equal chance. In this case, $\forall \mathbf{x}, P(\mathbf{x}) = 0.25$. Given the new formal definition in this paper, $P(\mathbf{x})$ is a problem dependent quantity and is independent from sample selection bias. Borrowing the notation of sample selection bias introduced in [2], a global classifier's dependence on sample selection bias is best formalized through the dependence on $P(s = 1|\mathbf{x})$ rather than on $P(\mathbf{x})$. In general, $P(s = 1|\mathbf{x})$ is not related to $P(\mathbf{x})$.

Definition 2.1 ***Improved and General Definition of Global Learners****: A global learner's output depends asymptotically on both $P(s = 1|\mathbf{x}, y)$ and $P(y|\mathbf{x})$. Under the second type of sample selection bias, it depends on $P(s = 1|\mathbf{x})$ and $P(y|\mathbf{x})$.*

In [2], $P(y|\mathbf{x})$ "refers to many local distributions, one for each value of $\mathbf{x}$". Strictly speaking, $P(y|\mathbf{x})$ denotes the true conditional probability distribution or the posterior probability distribution for a feature vector $\mathbf{x}$ to be a member of class y. The true class label for $\mathbf{x}$ is generated according to $P(y|\mathbf{x})$. $P(y|\mathbf{x})$ is completely determined by some unknown true function, and is unrelated to either the training data D or the model space of the inductive learner. Obviously, the true probability $P(y|\mathbf{x})$ is independent from both the sample selection bias due to training data D and the inductive bias due to choice of hypotheses space of a particular algorithm. In reality, for most practical applications where the dataset is not synthesized, the true probability distribution $P(y|\mathbf{x})$ is not known either before or after training some model. The availability of true probability $P(y|\mathbf{x})$ to model is very different from the true class label y. Class labels y's are provided in the training data, and are essential for inductive learners to construct classifiers. However, the true probability $P(y|\mathbf{x})$ is normally not given. Even if a feature vector $\mathbf{x}$ in the training data has class label y, it is strongly biased to assume that $P(y|\mathbf{x}) = 1$ in general. One instance of $(\mathbf{x}, y)$ is just a single observation. By definition, $P(y|\mathbf{x})$ is the probability to observe class label y when $\mathbf{x}$ is sampled repeatedly.

Classifiers as Approximators of $P(y|\mathbf{x})$: Most inductive learning algorithms construct classifiers to either directly or indirectly measure, approximate and output the true probability $P(y|\mathbf{x})$ by searching for one or a set of hypotheses that are consistent with the training data D in some hypotheses space Θ specific to each learning algorithm. The choice of hypothesis is called inductive bias. The model space for decision tree learning algorithm is the complete set of trees that tests each feature of the feature vector in different orders and with different splitting conditions. However, the model for logistic regression is the set of logistic regression formulas with different coefficients corresponding to each feature. For classification algorithms that do not directly output conditional probabilities, they either output a score (specific to each algorithm) or predict the "most likely" class label. The scores can be normalized into conditional probability estimates. When scoring is unavailable, we interpret the estimated probability for the predicted class as 1, and 0 for all others. In summary, classifiers can be represented as approximators of the true conditional probability. Formally, we use the notation $P(y|\mathbf{x}, D, \Theta)$ to denote a classifier constructed by some learner to approximate $P(y|\mathbf{x})$. In this notation, D is the training set of examples with $s = 1$ or $D = \{\forall(\mathbf{x}, y) \in \mathcal{D} \wedge s = 1\}$, Θ is the model space implied by the learning algorithm, such as the complete set of decisions trees. The learning algorithm searches for a model (or an ensemble of hypotheses) $\theta \in \Theta$ that is consistent with the training data D. When there are multiple hypotheses consistent with the training data D, preferences are given to certain models. We could use θ to replace the dependency on both D and Θ in the notation, i.e., we define $P(y|\mathbf{x}, \theta) = P(y|\mathbf{x}, D, \Theta)$. In our notation, θ is the "output of the learner" used in the original definitions of "local" and "global" or Definitions 1.1 and 1.2. The chosen model θ specifies the exact procedures to compute and appro

the true probability $P(y|\mathbf{x})$. If θ is a decision tree, it specifies the order of feature tests, the threshold value for continuous variable tests, and the number of examples belonging to different classes at the leaf nodes.

Learner Consistency at Estimating Probabilities: The classifier θ is trained to estimate/approximate the true probability $P(y|\mathbf{x})$. However, the estimated probability $P(y|\mathbf{x}, \theta)$ may not be equal to $P(y|\mathbf{x})$, if the true model or a model that produces $P(y|\mathbf{x})$ is not contained in the model space of the learner. We say that a learner is consistent if the learning algorithm can find a model θ that is equivalent to the true model at producing class conditional probabilities given an exhaustive training data set. Formally, a learner is consistent if it can find a model θ from an exhaustive number of examples such that $\forall \mathbf{x} \lim_{|D|\to\infty} P(y|\mathbf{x}, \theta) = P(y|\mathbf{x})$. For example, if a decision tree algorithm is used to approximate a non-vertical and non-horizontal linear function that separate the space into two classes, it will never be able to find a tree with 0 error rate. At best, a decision tree approximates the linear function with steps. Clearly, the consistency of a particular learner depends on the dataset that it is applied to. In other words, the same learner could be consistent for some dataset but inconsistent for others. Verification of learner consistency for an arbitrary dataset is a difficult problem. To the best of our knowledge, there is no published work to exhaustively test a learner's consistency for an arbitrary dataset. A complete answer is impossible for realistic problems with infinite number of examples and unknown true function, such as mortgage application and catalog campaigns. Based on the above analysis, we propose to relax Assumptions 1.2 and 1.3.

Assumption 2.1 ***Relaxed assumption of Assumption 1.2 and 1.3*** *We do not assume the learner's consistency, i.e., it could be either consistent or inconsistent.*

Limited Utility of $P(y|\mathbf{x}, s = 1) = P(y|\mathbf{x})$: Re-writing $P(s|\mathbf{x}, y) = P(s|\mathbf{x})$ by the definition of conditional probability, it becomes $\frac{P(s,y,\mathbf{x})}{P(y,\mathbf{x})} = \frac{P(s,\mathbf{x})}{P(\mathbf{x})}$. Re-arranging the denominators and using the definition of conditional probabilities, it becomes obvious that $P(y|s = 1, \mathbf{x}) = P(y|\mathbf{x})$. Following the definition of training dataset D, the dependency on $s = 1$ can be replaced by a dependency on the training set D, i.e., $P(y|D, \mathbf{x}) = P(y|\mathbf{x})$. $P(y|s = 1, \mathbf{x}) = P(y|\mathbf{x})$ is used in [2] to argue that some algorithms, e.g., Bayesian classifier, logistic regression and SVM, are independent from the second type of sample selection bias. However, this is only true under assumption 1.2 that the learner is consistent. However, as discussed in Section 2, the estimated probability is actually $P(y|\mathbf{x}, \theta)$, and we generally cannot assume the learners are consistent for many realistic problems. One deeper interpretation is that any consistent learner is "sufficiently and necessarily" local under the second type of sample selection bias since $\big(P(y|s = 1, \mathbf{x}) = P(y|\mathbf{x})\big) \Longleftrightarrow \big(P(s = 1|\mathbf{x}, y) = P(s = 1|y)\big)$.

Bayesian Classifiers: In the analysis of Bayesian classifiers to be "local" [2], the following equation is used $\frac{P(\mathbf{x}|y,s=1)P(y|s=1)}{P(\mathbf{x}|s=1)} = P(y|\mathbf{x}, s = 1) = P(y|\mathbf{x})$. This above analysis does not consider the dependency on the model space of the learner or Θ. For a Bayesian classifier, Θ describes exactly how to estimate $P(\mathbf{x}|y, s = 1)$ and $P(\mathbf{x}|s = 1)$ from the training data D with sample selection bias. In fact, $P(\mathbf{x}|y, s = 1)$ and $P(\mathbf{x}|s = 1)$ are $P(\mathbf{x}|y, s = 1, \Theta)$ and $P(\mathbf{x}|s = 1, \Theta)$ respectively. By definition of D, the dependency on $s = 1$ can be replaced by the dependency on D. These two probabilities can be then represented as $P(\mathbf{x}|y, D, \Theta)$ and $P(\mathbf{x}|D, \Theta)$ instead. As an example, suppose that we have a dataset with all categorical features. Obviously, $P(\mathbf{x}|y, D, \Theta)$ is the ratio of all examples in the training data D with class label y that also have feature vector $\mathbf{x}$. Now, the problem is to decide what numbers to divide to calculate this ratio. For simplicity, we assume that each feature vector $\mathbf{x}$ is unique. In this case, if the choice is to consider every feature x_i in the feature vector $\mathbf{x}$, then $P(\mathbf{x}|y, D, \Theta) = \frac{1}{|D_y|}$ if $\mathbf{x}$ has class label y, otherwise 0. In this notation, D_y is the subset of examples in the training dataset D that have class label y. $P(\mathbf{x}|D, \Theta)$ is $\frac{1}{|D|}$ because each feature vector $\mathbf{x}$ is assumed to be unique. Taking everything into consideration, $P(y|\mathbf{x}, D, \Theta) = 1$ if $\mathbf{x}$ has class label y or 0 otherwise. This computation is straightforward but not very useful, since there is no generalization and it is equivalent to "rote learning". The problems come from the strong assertions to consider every feature x_j in the feature vector $\mathbf{x}$. In reality, we normally only consider a "subset" of features in the feature vector $\mathbf{x}$ in order for the algorithm to generalize. The exact subset of features to consider depends on the training data D. Since Θ actually represents these inevitable choices, the dependency on Θ cannot be ignored. In summary, due to the inevitable assertions and choices, it is generally very hard to compute $P(y|\mathbf{x})$ exactly for Bayesian classifiers. As a matter of fact, we still compute $P(y|\mathbf{x}, D, \Theta)$, and neither the dependency on D nor the dependence on Θ can be removed. To be specific, we provide an example to show that Bayesian classifier can also be "global", as opposed to the analysis in [2] that it is always "local". Assume that the feature vector can be decomposed into two disjoint subsets $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2)$. Let the true conditional probability only depend on $\mathbf{x}_1$, i.e., $P(y|\mathbf{x}) = P(y|\mathbf{x}_1)$, and the selector variable depend only on $\mathbf{x}_2$, i.e., $P(s = 1|\mathbf{x}) = P(s = 1|\mathbf{x}_2)$. Under this situation, the choice of features to compute $P(\mathbf{x}|y, s = 1)$ and $P(\mathbf{x}|s = 1)$ decides how much the sample selection bias will influence these estimated quantities from the data. If luckily, only those features $\in \mathbf{x}_1$ are taken into account to compute $P(\mathbf{x}|y, s = 1)$ and $P(\mathbf{x}|s = 1)$, then the effect of sample selection bias will be rather small. However, if any features $\in \mathbf{x}_2$ are chosen to compute $P(\mathbf{x}|y, s = 1)$ and $P(\mathbf{x}|s = 1)$, the estimated probability will reflect sample selection bias.

Naive Bayes Classifier: In [2], naive Bayes is argued to a global classifier that is affected by sample selection bias since the independence assumption may not always hold true. However, this does not imply that the naive B

classifier is always a "global classifier" for any datasets. For some datasets where the independence assumption holds true, naive Bayes computes exactly $\frac{P(\mathbf{x}|y,s=1)P(y|s=1)}{P(\mathbf{x}|s=1)} = P(y|\mathbf{x}, s=1)$, which is $P(y|\mathbf{x})$ according to the assumption of the second type of sample selection bias. A simple example that satisfies the independence assumption is one whose true label is only dependent on one feature, and all other features are irrelevant, i.e., $\exists i, P(y|\mathbf{x}) = P(y|x_i)$.

Logistic Regression In [2], logistic regression is described as $P(y=1|\mathbf{x}, s=1) = \frac{1}{1+\exp(\beta_0+\beta_1 x_1+\ldots+\beta_n x_n)}$. Logistic regression is argued to be free from the second type of sample selection bias in [2] as "because we are assuming that y is independent of s given $\mathbf{x}$ we have that $P(y=1|\mathbf{x}, s=1) = P(y=1|\mathbf{x})$". This assumption ignores the effect of D on θ, or the set of parameters β_i in this case. A further proof would be required to justify that β_i's are not affected by $s=1$ for any dataset D. However, this could be very difficult since $\beta_i's$ are computed from D by minimizing log-likelihood function through Newton's method. In [2], a simple example of one independent variable is shown to justify the claim that logistic regression is not affected by sample selection bias. Although this example is indeed correct, it is over simplified. The true function for the given example is a simple linear separable function $P(1|x \geq -0.75) = 1$. The likelihood equation of logistic regression, both binary response and ordinal response (> 2 classes), is not guaranteed to have a finite solution. The existence of maximum likelihood estimate depends on the configuration of chosen training examples [1]. There are three mutually exclusive and exhaustive categories, i.e., complete separation, quasicomplete separation and overlap. For a formal and detailed explanation of these three cases, please refer to [1]. In summary, complete separation equals to the simplest case of "linear separatability". As long as all the data points in the domain $\mathbf{x}$ are linearly separatable, logistic regression is completely free from any sample selection bias including the second type of sample selection bias as discussed in [2]. However, in "quasicomplete" and "overlap", the data points are not "linearly" separable. In both cases, the solution , i.e, β's, are sensitive to the second type of sample selection bias. An illustrative example can be found in [1].

Decision Trees: Decision trees are argued to be "global" classifier independent from the dataset and problem in [2]. However, a decision tree could also be a "local" classifier. The decision path of a tree tests a sequence of features starting from the root of tree to the current node. Without loss of generality, assume that decision path is $(x_1, x_2, \ldots, x_k)$, which is a true subset of the full feature vector, or $\subset \mathbf{x} = (x_1, x_2, \ldots, x_n)$. Assume that each feature is categorical. Then $P(y|\mathbf{x}, s) = P(y|\mathbf{x})$ implies $P(y|x_1, x_2, \ldots, x_k, s) = P(y|x_1, x_2, \ldots, x_k)$ if $x_{k+1}, \ldots, x_n$ are irrelevant at predicting the class label y but may be exclusively used to determine if the instances are selected into the training set.

SVM: In [2], the hard-margin SVM algorithm is argued to be a local learner, while the soft-margin SVM algorithm is argued to be a global learner. The hard-margin SVM algorithms learn the parameters a and b describing a linear decision rule, $h(\mathbf{x}) = \text{sign}(a \cdot \mathbf{x} + b)$ whose sign determines the label of an example, so that the smallest distance between each training example and the decision boundary, i.e. the margin, is maximized. Given a sample of examples $(\mathbf{x_i}, y_i)$, where $y_i \in \{-1, 1\}$, it accomplishes margin maximization by solving the following optimization problem: minimize: $V(a, b) = \frac{1}{2}a \cdot a$, subject to: $\forall i: \ y_i[a \cdot \mathbf{x_i} + b] \geq 1$. The constraint requires that all examples in the training set are classified correctly, i.e., that the data can be separated by a hyperplane. If this is indeed the case, sample selection bias will not asymptotically affect the output of the hard-margin SVM algorithm as argued by [2]. However, because this algorithm does not have a solution if the data is not linearly separable it cannot be applied in most practical cases. The soft-margin SVM algorithms introduces slack variables $\xi_i > 0$ for each example $(\mathbf{x_i}, y_i)$. The optimization is changed to minimize: $V(a, b, \xi) = \frac{1}{2}a \cdot a + C\sum_{i=1}^{n} \xi_i$, subject to: $\forall i: \ y_i[a \cdot \mathbf{x_i} + b] \geq 1 - \xi_i, \ \xi_i > 0$ If a training example lies on the wrong side of the decision boundary, the corresponding ξ_i is greater than 1. Therefore, $\sum_{i=1}^{n} \xi_i$ is an upper bound on the number of training errors. The factor C is a parameter that allows one to trade off training error and model complexity. In [2] it is argued that sample selection bias affects the soft-margin SVM because it can change the sum of ξ_i values by making regions of the feature space denser than others. When this sum is changed, the decision boundary is also changed. Therefore, the soft-margin SVM algorithm is characterized as a global leaner. While this is true in general, like other learners, the soft-margin SVM algorithm can also behave as a local learner depending on the specific dataset used. In particular, if the data is linearly separable, the sum of ξ_i values will be always zero and sample selection bias will not asymptotically affect the output. This is also the case if the minimum of the sum is not changed by the bias (for example if the bias only affects examples that are on the correct side of the boundary).

3 Conclusion

One important contribution of this paper is to relax the assumptions made in an earlier paper [2] and argue formally and by examples that most classifier learners could be sensitive or insensitive to a common type of sample selection bias where the chance of an example being selected into the training set depends on its feature vector $\mathbf{x}$ but not directly on its class label. In conclusion, a learner could be sensitive or global [2] for some datasets and insensitive or local [2] for others. A longer version of this paper including experimental results verifying theoretical analysis and a detailed review of many related works is available upon request.

References

[1] Ying So. A tutorial on logistic regression. Technical report, SAS Institute Inc, Cary, NC, 1999.

[2] Bianca Zadrozny. Learning and evaluating classifiers under sample selection bias. In *Proceedings of the 21th International Conference on Machine Learning*, 2004.