Learning and Evaluating Classifiers under Sample Selection Bias

Bianca Zadrozny ZADROZNY@US.IBM.COM
IBM T.J. Watson Research Center, Yorktown Heights, NY 10598

Abstract

Classifier learning methods commonly assume that the training data consist of randomly drawn examples from the same distribution as the test examples about which the learned model is expected to make predictions. In many practical situations, however, this assumption is violated, in a problem known in econometrics as sample selection bias. In this paper, we formalize the sample selection bias problem in machine learning terms and study analytically and experimentally how a number of well-known classifier learning methods are affected by it. We also present a bias correction method that is particularly useful for classifier evaluation under sample selection bias.

1. Introduction

One of the most common assumptions in the design of learning algorithms is that the training data consist of examples drawn independently from the same underlying distribution as the examples about which the model is expected to make predictions. In many real-world applications, however, this assumption is violated because we do not have complete control over the data gathering process.

For example, suppose we are using a learning method to induce a model that predicts the side-effects of a treatment for a given patient. Because the treatment is not given randomly to individuals in the general population, the available examples are not a random sample from the population. Similarly, suppose we are learning a model to predict the presence/absence of an animal species given the characteristics of a geographical location. Since data gathering is easier in certain regions than others, we would expect to have more data about certain regions than others.

In both cases, even though the available examples are not a random sample from the true underlying distribution of examples, we would like to learn a predictor from the examples that is as accurate as possible for this distribution. Furthermore, we would like to be able to estimate its accuracy for the whole population using the available data.

This problem has received a great deal of attention in econometrics, where it is called sample selection bias. 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 procedure for correcting sample selection bias. 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, commonly used in econometrics.

Also, in statistics, the related problem of missing data has been considered (Little & 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.

In this paper, we address the sample selection bias problem in the context of learning and evaluating classifiers. In Section 2 we formally define the sample selection bias problem in machine learning terms. In Section 3 we present a new categorization of learning methods that is useful for characterizing their behavior under sample selection bias and study how a number of well-known classifier learning methods are affected by sample selection bias. In Section 4, we present a bias correction method based on estimating the probability that an example is selected into the sample and using rejection sampling to obtain unbiased samples of the correct distribution. It can be used both for learning classifiers and, more importantly, for evaluating a classifier using a biased sample.

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004. Copyright 2004 by the first author.

2. Definition

Standard classifier learning algorithms (implicitly or explicitly) assume that we have examples (x, y), each drawn independently from a distribution D with domain $\mathcal{X} \times \mathcal{Y}$ where $\mathcal{X}$ is the feature space and $\mathcal{Y}$ is a (discrete) label space.

Here, we assume that examples (x, y, s) are drawn independently from a distribution D with domain $\mathcal{X} \times \mathcal{Y} \times \mathcal{S}$ where X is the feature space, Y is the label space and S is a binary space. The variable s controls the selection of examples (1 means the example is selected, 0 means the example is not selected). We only have access to the examples that have $s = 1$, which we call the selected sample. If the selected sample (ignoring s) is not a random sample of D we say that the selected sample is biased.

There are four cases worth considering regarding the dependence of s on the example (x, y) [1]:

1. If s is independent of x and independent of y, the selected sample is not biased, that is, the examples (x, y, s) which have $s = 1$ constitute a random sample from D (ignoring s).

2. If s is independent of y given x (that is $P(s|x, y) = P(s|x)$), the selected sample is biased but the biasedness only depends on the feature vector x.

3. If s is independent of x given y (that is $P(s|x, y) = P(s|y)$), the selected sample is biased but the biasedness depends only on the label y. This corresponds to a change in the prior probabilities of the labels. This type of bias has been studied in machine learning literature and there are methods for correcting it (Elkan, 2001; Bishop, 1995).

4. If no independence assumption holds between x, y and s, the selected sample is biased and we cannot hope to learn a mapping from features to labels using the selected sample, unless we have access to an additional feature vector x_s that controls the selection (that is, $P(s|x_s, x, y) = P(s|x_s)$) for all the examples (even for the ones that have $s = 0$).

In econometrics, the usual assumption is (4) because the goal is to estimate the parameters of a model for y that reflects the true dependence of y on x. Any feature variable that only affects the selection should not be included in x (and it is included in x_s, instead). In classifier learning, this is not a concern, because we are mostly interested in the predictive performance of the model and not in making conclusions about the underlying mechanisms that generate the data.

For this reason, we argue that the most important sample selection bias case in the practice of classifier learning is case (2). In order to make the condition $P(s|x, y) = P(s|x)$ true in practice, the input to the classifier x has to include all the variables that affect the sample selection. For example, in the medical treatment case, we need to include in x the variables about the patients that the doctors use to decide who gets the treatment (even if they do not affect the side-effects of the treatment directly).

Even if this assumption is not true in practice (either because we do not have access to all the variables that control the selection or because it truly depends directly on y), assuming case (2) is more realistic than the usual assumption of case (1). In the rest of this paper, sample selection bias will refer to case (2).

3. Learning under sample selection bias

We can separate classifier learners into two categories:

- local: the output of the learner depends asymptotically only on $P(y|x)$
- global: the output of the learner depends asymptotically both on $P(x)$ and on $P(y|x)$.

The term "asymptotically" refers to the behavior of the learner as the number of training examples grows. The names "local" and "global" were chosen because $P(x)$ is a global distribution over the entire input space, while $P(y|x)$ refers to many local distributions, one for each value of x. Local learners are not affected by sample selection bias because, by definition $P(y|x, s = 1) = P(y|x)$ while global learners are affected because the bias changes $P(x)$.

Although this categorization is very simple, it is not straightforward to classify existing learners into it. Below, we study analytically and experimentally how sample selection bias affects different types of classifiers learning methods, including Bayesian classifiers, logistic regression, SVM and decision trees.

3.1. Bayesian classifiers

Bayesian classifiers compute posterior probabilities $P(y|x)$ using Bayes' rule:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)}$$

[1] In the statistics literature on missing data (Little & Rubin, 2002), cases (1), (2) and (4) are known as missing completely at random (MCAR), missing at random (MAR) and not missing at random (NMAR), respectively.

where $P(x|y)$, $P(y)$ and $P(x)$ are estimated from the training data. An example x is classified by choosing the label y with the highest posterior $P(y|x)$.

We can easily show that bayesian classifiers are not affected by sample selection bias. By using the biased sample as training data, we are effectively estimating $P(x|y, s = 1)$, $P(x|s = 1)$ and $P(y|s = 1)$ instead of estimating $P(x|y)$, $P(y)$ and $P(x)$. However, when we substitute these estimates into the equation above and apply Bayes' rule again, we see that we still obtain the desired posterior probability $P(y|x)$:

$$\frac{P(x|y, s = 1)P(y|s = 1)}{P(x|s = 1)} = P(y|x, s = 1) = P(y|x)$$

since we are assuming that y and s are independent given x. Note that even though the estimates of $P(x|y, s = 1)$, $P(x|s = 1)$ and $P(y|s = 1)$ are different from the estimates of $P(x|y)$, $P(x)$ and $P(y)$, the differences cancel out. Therefore, bayesian learners are local learners.

In practice, we have a limited amount of examples to estimate $P(y|x)$. Compared to a random sample of the same size, the biased sample contains more examples in parts of the feature space where $P(s = 1|x)$ is high and less examples where $P(s = 1|x)$ is low. This will lead to estimates of $P(y|x)$ with lower variance where $P(s = 1|x)$ is high and with higher variance where $P(s = 1|x)$ is low. However, as long as $P(s = 1|x)$ is greater than zero for all x, as we increase the sample size, the results on a selected sample will asymptotically approach the results on a random sample.

3.1.1. Naive Bayes

In practical Bayesian learning, we often make the assumption that the features are independent given the label y, that is, we assume that

$$P(x_1, x_2, \ldots, x_n|y) = P(x_1|y)P(x_2|y) \ldots P(x_n|y).$$

This is the so-called naive Bayes assumption.

With naive Bayes, unfortunately, the estimates of $P(y|x)$ obtained from the biased sample are incorrect. The posterior probability $P(y|x)$ is estimated as

$$\frac{P(x_1|y, s = 1) \ldots P(x_n|y, s = 1)P(y|s = 1)}{P(x|s = 1)},$$

which is different (even asymptotically) from the estimate of $P(y|x)$ obtained with naive Bayes without sample selection bias. We cannot simplify this further because there are no independence relationships between each x_i, y and s. Therefore, naive Bayes learners are global learners.

3.2. Logistic regression

In logistic regression, we use maximum likelihood to find the parameter vector β of the following model:

$$P(y = 1|x) = \frac{1}{1 + \exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}$$

With sample selection bias we will instead fit:

$$P(y = 1|x, s = 1) = \frac{1}{1 + \exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n)}$$

However, because we are assuming that y is independent of s given x we have that $P(y = 1|x, s = 1) = P(y = 1|x)$. Thus, logistic regression is not affected by sample selection bias, except for the fact that the number of examples is reduced. Asymptotically, as long as $P(s = 1|x)$ is greater than zero for all x, the results on a selected sample approach the results on a random sample. In fact, this is true for any learner that models $P(y|x)$ directly. These are all local learners.

Figure 1 illustrates the effect of sample selection bias on logistic regression for synthetically generated data, where x is one-dimensional. The graph on the left shows 1000 points where the x value is chosen uniformly between -10 and 10 and the y value is drawn with probabilities calculated using a logistic function (β_0=3 and β_1=2). The curve is the logistic function obtained using the plotted points. The dashed line is the separator between the two classes. The graph on the right shows a selected sample of the points, where the probability of each point being selected is proportional to its x value. We also show the logistic function obtained using the selected points. Although the selected sample contains many less points on the negative side than the original sample, the estimated curve and the resulting separator are the same.

3.3. Decision tree learners

Decision tree learners such as C4.5 (Quinlan, 1993) and CART (Breiman et al., 1984) split the input space x in a recursive, top-down manner. Each branch of the tree is a test on the value of one the features. For discrete features, the tree branches into nodes corresponding to each of the possible values. For real-valued features, the tree branches into two nodes corresponding to some threshold. To predict the class of a new example, we work down the tree, at each node choosing the appropriate branch by comparing the example with the values of the variable being tested for that node (Hand et al., 2001). The splitting criteria used by different decision tree learners vary, but they are all based on the nodes' impurity after the split. For example, CART uses the GINI index

$$\text{GINI(t)} = 1 - \sum_y P(y|t)$$

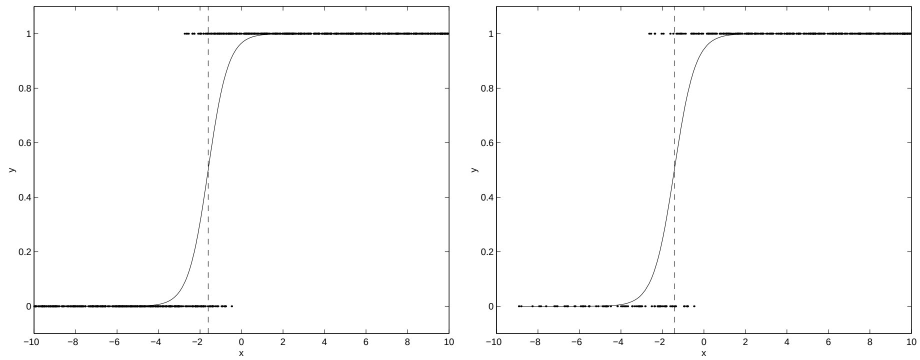

Figure 1. Logistic regression is unaffected by sample selection bias.

where $p(y|t)$ is the relative frequency of class y at node t. GINI is maximal when the examples are equally distributed among the classes and minimal when all the examples belong to one class. For each possible split, CART calculates $\sum_{i=1}^{k} \frac{n_i}{n}\text{GINI}(i)$, where n is the number of records at the node, n_i is the number of records at child i and k is the number of children induced.

C4.5 uses an information gain criterion given by

$$\text{INFO(t)} = -\sum_{y} P(y|t)\log P(y|t)$$

where $P(y|t)$ is the relative frequency of class y at node t. Like GINI, INFO is maximal when the examples are equally distributed among the classes and minimal when all the examples belong to one class.

Because the splitting criteria are dependent on $P(y|t)$, where t is a test on only one of the feature values, and, in general, $P(y|t, s = 1) \neq P(y|t)$, the splits chosen by the learners are sensitive to sample selection bias. Thus, decision tree learners are global learners.

3.4. Support vector machines

In its basic form, the support vector machine (SVM) algorithm (Joachims, 2000a) learns the parameters a and b describing a linear decision rule

$$h(x) = \text{sign}(a \cdot x + b),$$

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 (x_i, y_i), where $y_i \in \{-1, 1\}$, it accomplishes margin maximization by solving the following optimization problem:

$$\begin{array}{l}\text{minimize: } V(a,b) = \frac{1}{2} a \cdot a \\ \text{subject to: } \forall i: \quad y_i[a \cdot x_i + b] \geq 1\end{array}$$

The constraint requires that all examples in the training set are classified correctly. Thus, sample selection bias will not systematically affect the output of this optimization, assuming that the selection probability $P(s = 1|x)$ is greater than zero for all x.

Figure 2 illustrates the effect of sample selection bias on SVM for synthetically generated data, where x is two-dimensional. The graph on the left-hand side shows 500 points for each of two classes, generated from two different gaussians. The line is the maximal margin separator. The graph on the right-hand side shows a selected sample from these points where the probability of each point being selected is proportional to its horizontal coordinate. We also show maximal marginal separator using the selected points. Although the selected sample contains many less points on the negative side than the original sample, the resulting separator is not significantly altered.

In practice, a decision rule that classifies all the examples correctly may not exist because of class overlap. To allow for misclassified examples, one introduces slack variables $\xi_i > 0$ for each example (x_i, y_i). This is called a soft margin SVM classifier (Schölkopf & Smola, 2002). The optimization is changed to

$$\begin{array}{l}\text{minimize: } V(a,b,\xi) = \frac{1}{2} a \cdot a + C \sum_{i=1}^{n} \xi_i \\ \text{subject to: } \forall i: \quad y_i[a \cdot x_i + b] \geq 1 - \xi_i, \ \xi_i > 0\end{array}$$

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. We note that the algorithm can be generalized to non-linear decision rules by replacing inner products with a kernel function (Joachims, 2000a).

While sample selection bias does not affect the hard margin SVM, it does affect the soft margin version because it considers the sum of ξ_i values. By making regions of the feature space denser than others, sample

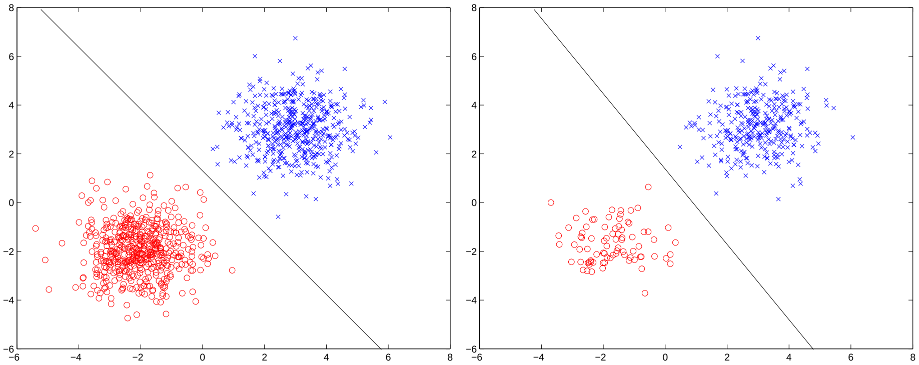

Figure 2. SVM for separable data is unaffected by sample selection bias.

selection bias changes this sum and, with it, the decision boundary. Soft margin SVM is a global algorithm because changes in $P(x)$ will change the output.

3.5. Experimental results

To verify the effects of sample selection bias experimentally, we apply Naive Bayes, logistic regression, C4.5 and SVMLight (soft margin) (Joachims, 2000b) to the Adult dataset, available from the UCI Machine Learning repository (Blake & Merz, 1998). We assume that the original dataset is not biased and artificially simulate biasedness by generating a value for s for each example, such that s is correlated with one of the input features. When training, we only use the examples in the training set for which $s = 1$. When testing, we use all the examples in the test set, because we are interested in measuring the performance of the classifiers on the original distribution of examples.

Figure 3 shows the results of applying the learners to the Adult dataset using unbiased and biased training sets of increasing size. For each size shown on the x-axis, we generated 50 unbiased samples from the original Adult training set. We also generated 50 biased samples by assigning s such that examples with feature `age` less than 30 are 9 times more likely to have $s = 1$ than examples with `age` more than 30. We trained the learners using each of the 50 samples (in both the biased and unbiased cases) and tested the models on the Adult test set, to obtain the mean and standard error of the error rate, as shown in the graphs.

In accordance with our analysis, for logistic regression, the difference in error rate between using a biased or an unbiased sample goes down as we increase the size of the training set. Also, as expected, we see that naive Bayes is very sensitive to sample selection bias. The error rate using the biased sample goes up as we increase the number of training examples.

Surprisingly, C4.5 performs very well under sample selection bias. This might be explained by the fact that even though the choice of splits is biased, the class estimates at the leaves are not. More research specific to decision tree learners is necessary to understand the effect of sample selection bias on them.

With SVM, we see that the error rate using the biased training set decreases as the training set sizes increases. However, the difference between the error rates using biased and unbiased samples does not decrease. This indicates that, asymptotically, SVM (with soft margin) is affected by sample selection bias.

4. Correcting sample selection bias

In the last section, we saw that some classifier learning methods are affected by sample selection bias, while others are not. In this section, we present a bias correction method that can be applied to any classifier learner, provided that we have a model for the selection probabilities $P(s = 1|x)$. The method works by correcting the distribution of examples through resampling and then applying the classifier learner to the corrected sample. It bears resemblance to weighting methods proposed in the statistics literature for missing data (Little & Rubin, 2002) and also to costing, a cost-sensitive learning method by example weighting presented in Zadrozny et al. (2003).

Classifier learners try to find h to minimize the expected value of loss function over the distribution of examples given by

$$E_{x,y \sim D}[l(h(x), y)].$$

The loss function is, in many cases, given by an indicator of error $I(h(x) \neq y)$, but we make the analysis more general by considering an arbitrary loss function.

Under sample selection bias, a classifier learner will

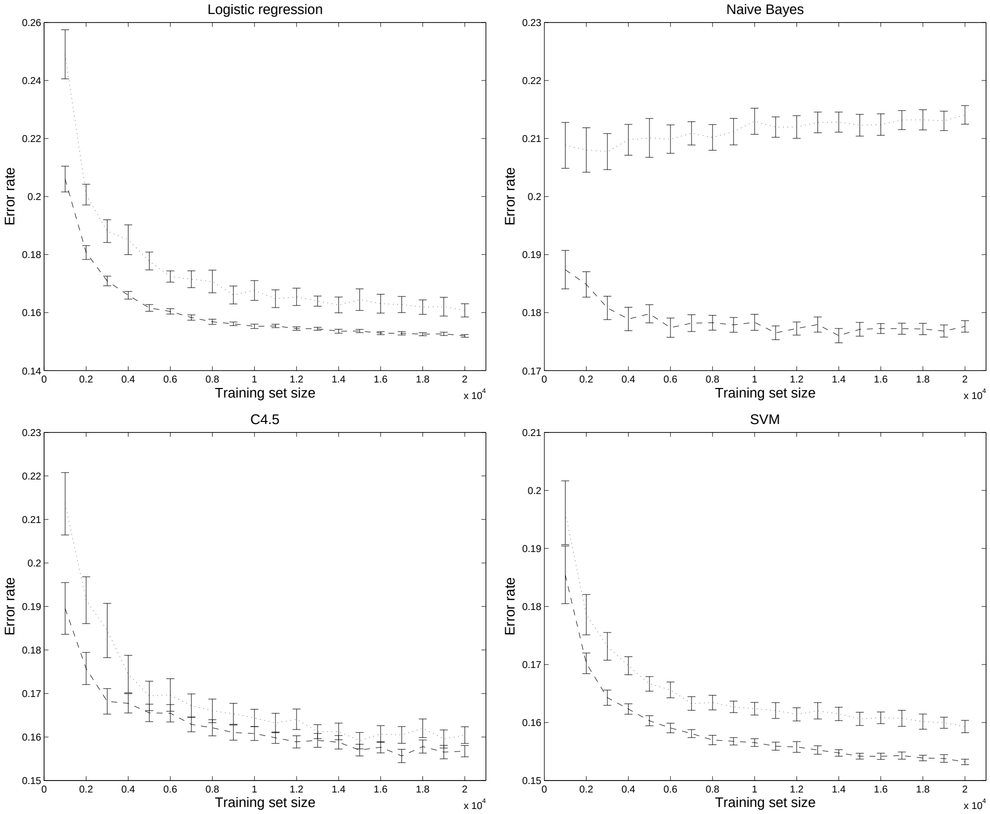

Figure 3. Error rate using biased (dotted) and unbiased (dashed) training sets. Each point shows the mean error rate for a given sample size and the bars show the standard error, computed using 50 different training sets for each size.

minimize instead

$$E_{x,y,s \sim D}[l(h(x), y)|s = 1]$$

because only the examples with $s = 1$ are available.

Assume that we know the selection probabilities $P(s = 1|x)$ and that they are greater than zero for all x. Let $\hat{D}$ be a new distribution such that

$$\hat{D}(x, y, s) \equiv P(s = 1)\frac{D(x, y, s)}{P(s = 1|x)}.$$

where $P(s = 1) = \sum_{(x,y,s) \sim D} P(s = 1, x)$ is the overall selection probability.

The following theorem shows that if we change the distribution of examples from D to $\hat{D}$, we will obtain the desired expected value under sample selection bias.

Theorem 1 *(Bias Correction Theorem) For all distributions, D, for all classifiers, h, for any loss function $l = l(h(x), y)$, if we assume that $P(s = 1|x, y) = P(s = 1|x)$ (that is, s and y are independent given x) then*

$$E_{x,y \sim D}[l(h(x), y)] = E_{x,y,\sim \hat{D}}[l(h(x), y)|s = 1]$$

Proof:

$$\begin{aligned}
&E_{x,y,s \sim \hat{D}}[l(h(x), y)|s = 1] \\
&= \textstyle\sum_{x,y} l(h(x), y) P_{\hat{D}}(x, y|s = 1) \\
&= \textstyle\sum_{x,y} l(h(x), y) \frac{P_D(s=1)}{P_D(s=1|x)} P_D(x, y|s = 1) \\
&= \textstyle\sum_{x,y} l(h(x), y) \frac{P_D(s=1)}{P_D(s=1|x)} \frac{P_D(s=1|x,y) P_D(x,y)}{P_D(s=1)} \\
&= \textstyle\sum_{x,y} l(h(x), y) P_D(x, y) \\
\blacksquare\ &= E_{x,y \sim D}[l(h(x), y)]
\end{aligned}$$

The left-hand side ($E_{x,y \sim D}[l(h(x), y)]$) is the expected value that we would like to minimize but cannot directly under sample selection bias. The right-hand side ($E_{x,y,s \sim \hat{D}}[l(h(x), y)|s = 1]$) can be minimized as long as we can draw examples from $\hat{D}$.

As discussed in Zadrozny et al. (2003), obtaining a sample from a weighted distribution given a finite set of training examples is not completely straightforward. They have demonstrated that costing, a method based on rejection sampling, achieves the best results in practice. For this reason, we recommend using costing for sample selection bias correction, where instead of using costs as weights we use the selection ratio $P(s = 1)/P(s = 1|x)$ as a weight for each example.

Up to now, we have assumed that we know the selection probabilities $P(s = 1|x)$. In practice, we would have to estimate these from data. If we have a sample $(x, s) \sim D$ (note that y is not necessary), we can use it to estimate these probabilities by feeding the sample to a classifier learner that outputs class membership probability estimates (using s as the label). Obtaining this sample is not difficult in many practical situations. For example, in medical treatment, we only know the outcome of the treatment (y) for patients x that were given the treatment ($s = 1$). On the other hand, we can obtain examples of the form (x, s) that are drawn from the population as a whole.

4.1. Evaluation under sample selection bias

In evaluation, for a given a classifier h, we want to obtain an estimate of the classifier loss, given by

$$E_{x,y\sim D}[l(h(x), y)].$$

Usually this is done by applying the classifier to a set of test examples drawn from D and obtaining the empirical loss on the test examples: $\frac{1}{m}\sum_{(x,y)} l(h(x), y)$, where m is the number of available examples.

Under sample selection bias, we only see the examples for which $s = 1$, and instead obtain an estimate of

$$E_{x,y,s\sim D}[l(h(x), y)|s = 1],$$

which generally is not an unbiased estimate of the loss.

As seen in Section 3, local learning methods are insensitive to sample selection bias. However, the evaluation step is always affected by sample selection bias because we are calculating an expected value over the whole input space (which is always "global"). Therefore, we argue that accounting for sample selection bias on the evaluation step is more important than accounting for sample selection bias during learning.

We can use the bias correction theorem for evaluating a classifier if we have estimates of the selection probabilities $P(s = 1|x)$. We simply have to weigh each example by $P(s = 1)/P(s = 1|x)$ when calculating the expected loss on the biased test sample. The unbiased empirical estimate of the loss should be calculated as

$$\frac{1}{m}\sum_{(x,y,s)} \frac{P(s=1)}{P(s=1|x)} l(h(x), y) = P(s = 1) \sum_{(x,y,s)} \frac{l(h(x), y)}{P(s = 1|x)}.$$

4.2. Example

To illustrate how the bias correction method works, we constructed an example using the KDD-98 competition dataset, available from the UCI KDD Archive (Bay, 2000). We assume we know the selection probabilities $P(s = 1|x)$ and we enforce the selection of examples using these probabilities. By doing this, we can compare the estimates of the expectation obtained using the whole sample and using the selected sample.

The KDD-98 dataset contains information about people who have made donations to a charity. For the purpose of this example, we only need to look at two variables: income and amount. Income indicates the person's income level and takes values in $\{0, 1, 2, 3, 4, 5, 6, 7\}$. Amount is the donation amount in the last campaign. We only use examples of people who have donated in the last campaign. In the notation of the theorem, income is x and amount is l. (We chose to side-step $h(x)$ and y and assume we have l directly for each example). Let s be such that

$$P(S = 1|X = x) = \begin{cases} 0.3 & \text{if } x \in \{0, 1, 2, 3\} \\ 0.9 & \text{if } x \in \{4, 5, 6, 7\} \end{cases}$$

where the overall selection probability $P(s = 1)$ is 0.6.

The empirical estimate of the expected amount obtained by averaging the amounts of all the examples is 15.62. Because there is a positive correlation between income and donation amount, if we select the examples according to the probabilities above, we will overestimate the expected amount.

To demonstrate this experimentally we can assign s values for each example according to the probabilities above and calculate the empirical mean of l using only the examples that have $s = 1$. By repeating this for 1000 different random draws of s, i.e., 1000 different selected samples, we obtain the distribution of estimated expected values of Y seen in Figure 4 (left). The vertical dashed line (on the left side) shows the estimated expected amount using the whole sample. The graph shows that, by using only the selected examples to estimate the expected value of l, we consistently overestimate it, as expected.

In contrast, Figure 4 (right) shows the distribution of estimated expected values for l, when we use only the selected examples but apply the bias correction method. The distribution is centered near the value estimated from the whole sample (and the mean is 15.62). Therefore, we can conclude that the proposed method succeeds at correcting the bias. We note, however, that there is a slight increase in variance.

5. Conclusions

We presented a formal definition of sample selection bias in classifier learning. By studying the behavior of different classifier learners under sample selection bias, we separated them into two categories:

- **local:** the asymptotical behavior only depends on $P(y|x)$. Examples: logistic regression, hard margin SVM.

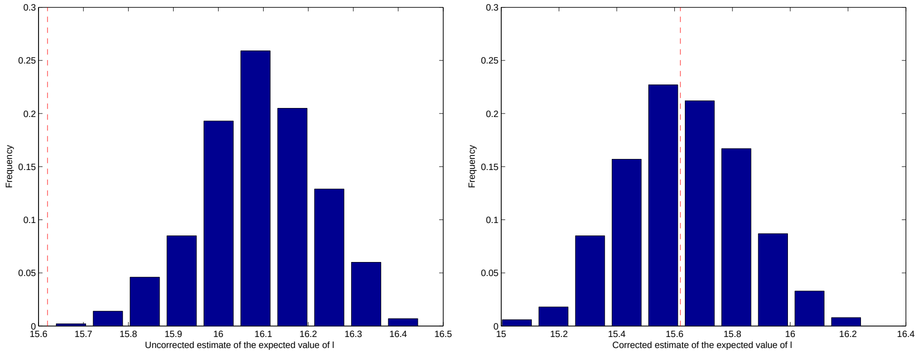

Figure 4. Distribution of the **uncorrected** (left) and **corrected** (right) estimates of expected amount (l) when different selected samples are used. The vertical dashed line shows the estimated expected value using the whole sample.

- **global:** the asymptotical behavior depends on both $P(x)$ and $P(y|x)$. Examples: naive Bayes, soft margin SVM, decision tree learners.

While global learners are affected by sample selection bias, local learners are not. This is a new categorization, different from the more usual categorization of learning methods into discriminative and generative (Ng & Jordan, 2002). As seen in Section 3.1, although generative (or Bayesian) methods model $P(x|y)$, $P(y)$ and $P(x)$, their behavior is generally independent of $P(x)$ (although this is not true for naive Bayes).

This categorization is also useful for defining situations in which we can learn from both labeled and unlabeled data, an area of research that has received some attention in recent years (see, for example, Szummer and Jaakkola (2003)). Clearly, global learners can take advantage of unlabeled data, while local learners cannot.

For global learners, we showed that we can still learn correctly under sample selection bias if we have data to estimate the selection probabilities $P(s = 1|x)$. Also, we showed how to evaluate a classifier using a biased sample and estimates of the selection probabilities.

Acknowledgments

I thank my Ph.D. advisor, Charles Elkan, for introducing me to sample selection bias and for reviewing earlier versions of this paper. I also thank the anonymous reviewers for their valuable suggestions.

References

Bay, S. D. (2000). UCI KDD archive. Department of Information and Computer Sciences, University of California, Irvine. `http://kdd.ics.uci.edu/`.

Bishop, C. (1995). *Neural networks for pattern recognition*, chapter 2. Oxford, UK: Clarendon Press.

Blake, C. L., & Merz, C. J. (1998). UCI repository of machine learning databases. Department of Information and Computer Sciences, University of California, Irvine. `http://www.ics.uci.edu/∼mlearn/MLRepository.html`.

Breiman, L., Friedman, J. H., Olsen, R. A., & Stone, C. J. (1984). *Classification and regression trees.* Wadsworth.

Elkan, C. (2001). The foundations of cost-sensitive learning. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence* (pp. 973–978).

Hand, D., Mannila, H., & Smyth, P. (2001). *Principles of data mining*, chapter 10.5. MIT Press.

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

Joachims, T. (2000a). Estimating the generalization performance of a SVM efficiently. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 431–438).

Joachims, T. (2000b). Making large-scale SVM learning practical. In B. Schölkopf, C. Burges and A. Smola (Eds.), *Advances in kernel methods - support vector learning.* MIT Press.

Little, R., & Rubin, D. (2002). *Statistical analysis with missing data.* Wiley. 2nd edition.

Ng, A., & Jordan, M. (2002). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. *Neural Information Processing Systems 14.*

Quinlan, J. (1993). *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels: Support vector machines, regularization, optimization, and beyond.* MIT Press.

Szummer, M., & Jaakkola, T. (2003). Information regularization with partially labeled data. *Neural Information Processing Systems 15.*

Zadrozny, B., Langford, J., & Abe, N. (2003). Cost-sensitive learning by cost-proportionate example weighting. *Proceedings of the Third IEEE International Conference on Data Mining* (pp. 435–442).