A Semi-supervised Approach for Reject Inference in Credit Scoring Using SVMs

Conference Paper · July 2010

DOI: 10.1007/978-3-642-14400-4_43 · Source: DBLP

Sebastián Maldonado
University of the Andes (Chile)
57 PUBLICATIONS **567** CITATIONS

SEE PROFILE

Gonzalo Paredes
University of Chile
11 PUBLICATIONS **16** CITATIONS

SEE PROFILE

A Semi-Supervised Approach for Reject Inference in Credit Scoring using SVMs*

Sebastián Maldonado‡ and Gonzalo Paredes †
‡Universidad de Los Andes, San Carlos de Apoquindo 2200,
Las Condes, Santiago, Chile.
†Department of Electrical Engineering, Universidad de Chile.
smaldonado@uandes.cl,goparede@ing.uchile.cl

Final version: July, 2010.

Abstract

This paper presents a novel semi-supervised approach that determines a linear predictor using Support Vector Machines (SVMs) and incorporating the information of rejected loans, assuming that the labeled data (accepted applicants) and unlabeled data (rejected applicants) are not drawn from the same distribution. We use a self training algorithm in order to predict how likely a rejected applicant would have repaid if the applicant would have received credit. A modification to the self training algorithm based on Platt's probabilistic output for SVMs is introduced. We consider two cases: when labeled and unlabeled data are obtained randomly but considering a higher risk in the unlabeled data, and when the decision of granting a loan is given by a set of rules and therefore the labeled and unlabeled data are not overlapped. Experiments with two toy data sets, one well-known benchmark Credit Scoring data set and one project performed for a Chilean financial institution demonstrates that our approach accomplishes the best classification performance compared to well-known reject inference alternatives and other state-of-art semi-supervised method for SVMs (Transductive SVM).

1 Introduction

Credit scoring represents sophisticated models to assess the risk of providing loan to a person or a business, rejecting those who are considered too risky.

[*] The present work was published in the Springer's Lecture Notes in Computer Science series, available at www.springer.com/lncs. Please cite this paper as follows: Maldonado, S., Paredes, G. A Semi-supervised Approach for Reject Inference in Credit Scoring Using SVMs. Advances in Data Mining. Applications and Theoretical Aspects. Lecture Notes in Computer Science, Volume 6171/2010, 2010, 558-571. DOI:http://dx.doi.org/10.1007/978-3-642-14400-4_43

Credit scoring models are used by all major banks and financial institutions because of their advantages: it significantly reduces loan processing costs and diminishes aggregate default costs [3].

Credit Scoring models are usually developed from granted loans (*known good/bad sample*), because complete data are only available for those accepted. However, a representative sample should be drawn from the population which applies for credit. Using a model based on only previously approved applicants can be inaccurate [26]. Furthermore, if the previous accept/decline decision was taken systematically, the set of accepted loans is a biased sample and not representative of the rejects (*sample bias*). A method is needed to account for cases where the behavior is unknown. Reject inference is therefore used to infer the status of applicants who have been rejected. [9].

The logit model is considered the main classification model in Credit Scoring [28]. Nevertheless, several data mining approaches have been proposed for this task [10]. The main objective of this work is to incorporate data mining techniques such as semi-supervised learning and SVMs to Credit Scoring in order to improve classification performance, considering a biased sample of the applicants. Another objective is to compare the performance of different reject inference approaches mentioned in the literature, together with semi-supervised methods such as self learning and transductive learning.

This paper is organized as follows. In Section 2 we briefly introduce semi-supervised learning for classification. Section 3 addresses the issue of non-random sample selection in Credit Scoring and the advantage of reject inference. Section 4 provides an overview on recent developments for reject inference in Credit Scoring. Section 5 introduces the proposed semi-supervised method based on SVM. Experimental results using two artificial and two real-world data set are given in Section 6. Section 7 summarizes this paper by providing its main conclusions and addresses future developments.

2 Semi-Supervised Learning

Semi-Supervised Learning (SSL) is a technique between supervised and unsupervised learning and it looks forward to make a better classification adding unlabeled data. Based on the fact that obtaining labeled data is expensive or difficult, being unlabeled data cheaper to obtain in many applications [7], SSL attempts to achieve a better classification performance using both labeled and unlabeled data. One of the first algorithms proposed for using unlabeled data is the self-training method [1, 25]. Another two important approaches are the co-training [4] and the Transductive SVM or S^3VM [17].

In the later sixties, transductive inference mixed with combinatorial optimization was applied by Hartley and Rao [15] in order to maximize the likelihood of their model. In the early seventies, semi-supervised learning appeared as a solution for Fisher linear discriminant with unlabeled data. The semi-supervised learning had been also applied to more theoretical analysis in the eighties and nineties, for example, learning rates in a probably approximately correct framework (PAC) by Valiant [29] and identifiable mixture, where Castelli and Cover [5] showed that with finite unlabeled points the probability of error has an exponential convergence to the Bayes risk.

In the nineties the interest in SSL increased thanks to text classification tasks. Nowadays semi-supervised learning is particularly important machine learning areas such as speech recognition, web mining and three-dimensional protein sequences problems [7]. The main algorithms of semi-supervised learning will be reviewed in the following sections.

2.1 Self-training

Self training, also known as well as self-labeling or decision-directed learning, is the most common and simple SSL method. This wrapper algorithm uses the prediction of a supervised learning method to label the unlabeled data. In other words, the classifier uses its own prediction to teach itself. It starts training a classifier only with labeled data. In each step the algorithm selects a fraction of the unlabeled examples for labeling, according to a target or a decision function. Then the method adds these objects to the training set. Finally the classifier retrains and the process is repeated.

The self-learning algorithm is very simple and can be used as a meta-learning algorithm. Nevertheless, it relays on the goodness of fit of the classifier obtained, considering that mistakes reinforce themselves. Other disadvantage of self-learning is the difficulty to analyze it in general, however for specific base learners there has been some studies of convergence [12, 14].

This method has been used in several natural language processing tasks like: word sense disambiguation [33], identification of subjective nouns [19] and machine translation. Another area in which self-training has been successfully applied is object detection systems for images [24]. This work shows that semi-supervised techniques work as good as state of the art detectors.

The Self-training will be one of the semi-supervised strategies that we will use in order to improved credit scoring models.

2.2 Co-training

Co-training methods are based on three assumptions. Firstly, there should be a natural split of variables in two subsets. Secondly, each subset should be sufficient in order to train a good classifier. Finally, the method assumes that both subsets are conditionally independent given the class.

The approach trains two different classifiers, one for each subset, using only the labeled data. Then each classification function classifies part of the unlabeled data and teaches the other classifier. Both classifiers are retrained with this new labeled data given by the other classifier (cross information) in an iterative way.

Nigam and Ghani [22] compare co-training with generative models and Expectation-Maximization (EM) algorithm. Their results show that co-training performs well, when the assumption of conditional independence is held. They also realized that it is better to perform probabilistic labeling of the entire universe rather than considering only the most confident unlabeled data. This work also states that if there is no natural feature split of the set, they could create an artificial split by randomly dividing the feature set in two. Although this artificial split helps, the results are not as good as in the case where the split is natural.

Other versions of co-training have been developed by relaxing the assumption on splitting features. Goldman and Zhou [13] consider two learners of different types, both using the whole feature set. One of these learners has a high confidence (based on statistical tests) and it is used to teach the other one and vice versa. This approach was proved for the case were labeled and unlabeled data do not follow the same distribution [8].

Democratic co-training is another example of an algorithm derived from original co-training [34]. In this case a group of learners with different inductive bias is trained separately, using the complete feature set from the labeled data. The approach classifies unlabeled data by agreement on the class of unlabeled examples and a variant of a weighted majority vote among all learners it is used to make the final prediction. Balcan et al. [2] and then Johnson and Zhang [18] have also relaxed the assumption of conditional independence in their works.

2.3 Transductive Support Vector Machine (TSVM) or S^3VM

Transductive Support Vector Machine is an extension of standard SVM, where only labeled data is used. The goal of TSVM is to use both labeled and unlabeled data in order to obtain the maximum margin in the linear boundary of the Reproducing Kernel Hilbert Space. Finding the exact TSVM solution is NP-hard, so great effort has been made on approximation algorithms. One of the first widely used software for solving this problem is SVMlight, TSVM implementation by Joachims [17]. Other approaches attempt to relax the TSVM training problem to a semi-definite programming (SDP), but this method still has a expensive computational cost. The optimization problem associated to TSVM is presented next.

Let $\varphi(\chi_i) = h(\chi_i) + b$ were $h \in H_K$.

$$\underset{\varphi}{\text{Min}} \quad \sum_{i=1}^{l} (1 - y_i \varphi(\chi_i)) + \lambda_1 \|h\|^2{}_{Hk} + \lambda_2 \sum_{i=l+1}^{n} (1 - |\varphi(\chi_i)|) \tag{1}$$

The last term of the equation takes into account the unlabeled data. The loss function $(1-|\varphi(\chi_i)|)$ has a non-convex hat shape, which is the source of difficulty in the optimization problem. Some researchers proposed to solve the optimization problem using Gaussian function as an approximation of the hat loss function [18]. Other approaches attempts to solve an easier problem and then gradually deform it into the TSVM objective. Collobert et al. [6] optimized the hard TSVM directly using an approximate optimization procedure call concave-convex procedure (CCCP). As a result the authors report improvements in speed for the training of the TSVM. A global optimal solution of

TSVM was proposed using Branch and Bound, where an excellent accuracy for small data sets is shown. Although Branch and Bound probably will not be useful for large data sets, this result shows the potential of TSVM with better approximation algorithms.

3 Reject Inference for Credit Scoring

Credit Scoring models are developed to predict the behavior of all applicants and using a model based only on approved ones can be inaccurate. This is a major issue when the accept/decline decisions are made systematically and not random. In this case the accepted population is not representative of the rejected loans and a method is needed for cases where the behavior is unknown. Reject inference is a process that forecast the behavior of reject applicants based on the analysis and performance of previously rejected. The main reason for performing reject inference is the sample bias issue.

Another important reason to make reject inference is the business relevance [26]. If we assume that there are some bads in the population that is approved, but also there will be some goods that have been declined. Reject inference could give to the credit grantor the tools to make a better and more informed decision making, approving the same number of people but obtain better results by better selection. One of the main goals of the credit grantor is to identify the so-called swap set. This set is basically the exchange of known bads with inferred goods that were rejected previously, but have been identified as potential goods using reject inference.

Reject inference can neutralize some distortions in decision-making. For example, if a credit is given to a group of applicants who have historic delinquency, and they respond as good applicants, a credit scoring models (without reject inference) could probably classify a new applicant who has historic delinquency as good, based on the result of the first group. This kind of distortions could be treated with reject inference. It is also useful to estimate level of risk in a specific unknown situation, allowing to estimate bad rates by the score of those who were previously rejected, helping decision-making processes. However, reject inference involves predicting an unknown, and will always have a degree of uncertainty. The level of uncertainty can be reduced using better techniques but it will never be 100% accurate.

Depending on the application acceptance rate and the level of confidence in previous credit-granting criteria, reject inference could have more impact on the credit scoring models. For example, with a very high level of confidence and a high approval rate reject inference is less important. In this case all rejected can be seen as bad with high level of confidence. If the level of confidence in the credit-granting criteria is very low it can be assume near random adjudication and again the reject inference it is not relevant. In cases with low or medium approval rates and low bad rates, reject inference helps to identify chances to increase market share with risk-adjusted strategies. Reject inference will also have an important impact in cases where the accept/decline decision process performs good.

Several strategies for reject inference in Credit Scoring have been proposed [26]. Subsection 3.1 summarizes different approaches for traditional reject inference. With the advances in data mining in the last decade, some strategies have been developed for Credit Scoring and reject inference using data mining techniques, which we present in subsection 3.2.

3.1 Traditional Reject Inference Strategies

There are various techniques used to perform reject inference. Some of these are presented in the next paragraphs.

Assign All rejects to bads. This approach is not adequate in most cases, because we know that an important fraction of the rejects would have been good. The only situation where this assumption could be suitable is when the approval rates are very high and the cost of default is very high as well.

Assign rejects in the same proportion of goods to bads as reflected in the accepted loans. We can use this assignation with confidence in two situations. If there is no consistency in the current selection system or if the decisions have been made randomly.

Ignore the rejects altogether. This is the most common method. The scoring system is developed only with accepted applicants and the sample bias issue is present. Ignoring the rejects is an ineffective and inefficient alternative.

Approve All Applications for a time period. This is the only method who allows to find out the actual or real performance of rejected accounts. It necessary to approved all applications for a specific time period, which could be a very expensive decision in terms of credit risk. The approved applications should be representative of all score ranges, so it is not acceptable to understate or overstate the bad rate of the rejects.

Use Data mining techniques to classify rejects. Based on the idea of rejected and approved applicants have different distribution, we can improve the classification performance by applying data mining techniques in order to incorporate the whole information of the applicants for prediction.

3.2 Reject Inference using Data Mining Techniques

Some approaches have been developed for reject inference for Credit Scoring. This issue has been mainly addressed in the context of logistic regression. Chen [9] proposed a maximum likelihood approach for reject inference, but it is limited to logistic regression. More general approaches such as Heckman's bivariate two-stage model and the augmentation method have been also proposed. Unfortunately, empirical research of these models shows little promise [9].

The issue of a non-random sample for the unlabeled data in semi-supervised learning has been addressed in a more general context (see, for example, [35]) and in different applications, such as spam filtering [32]. An interesting approach based on bayesian networks is proposed for biased labeling in semi-supervised learning.

Transductive SVM also allows to perform reject inference by an additional parameter p, which represents the fraction of unlabeled examples to be classified into the positive class [17]. This parameter allows a correction to the classifier but does not perform reject inference since this method does not classify the unlabeled data.

4 Semi-Supervised Algorithm for Reject Inference in Credit Scoring

We propose a self-training algorithm with a modification in order to incorporate the assumption that the unlabeled data (rejected loans) have a higher risk in terms of good/bad proportion. The main idea is to train a SVM classifier using the labeled data (accepted applicants) and to estimate the probability of default for rejected loans by using a logit link function as proposed by Platt [23]. The next step is to adjust the cut-off threshold using a parameter λ and computing the confidence of each unlabeled object. We iteratively incorporate the unlabeled observations with higher confidence to the labeled data until all unlabeled data is labeled. The final classifier consider all applicants for credit and allows an unbiased prediction for the behavior of new applicants in terms of risk.

The intuition behind this approach is that we can adjust the classifier by penalizing the unlabeled objects with less confidence (closer to the hyperplane), forcing some of the rejected loans labeled as ”good” in the self training process to be ”bad”. Unlabeled objects with high confidence are more likely to be consistent with the classifier, so we focus in modify the classifier using the unlabeled objects with less confidence in order to incorporate the higher risk of rejected loans and to construct a classifier based on an unbiased sample of the real population of applicants.

Formally, given training vectors $\mathbf{x}_i \in \Re^M$, $i = 1, ..., N$, which consists in L labeled examples $\left\{(\mathbf{x_i^l}, \mathbf{y_i^l})\right\} \in \mathscr{L}$, $y_i^l \in \{-1, +1\}$, and U unlabeled examples $\{\mathbf{x_i^u}\} \in \mathscr{U}$, where $N = L + U$ and $\mathscr{L}$ and $\mathscr{U}$ represent the sets of labeled and unlabeled examples respectively. For binary classification, SVM provides the optimal hyperplane $f(\mathbf{x}) = \mathbf{w^T} \cdot \mathbf{x^l} + \mathbf{b}$ that aims to separate the training patterns. In the case of linearly separable classes this hyperplane maximizes the sum of the distances to the closest positive and negative training patterns. This sum is called *margin*. To construct the maximum margin or optimal separating hyperplane, we need to classify correctly the vectors $\mathbf{x}_i^l$ of the training set into two different classes y_i^l, using the smallest norm of coefficients $\mathbf{w}$ [30].

If we look for a linear hyperplane in the case of linearly non-separable classes, a set of slack variables is introduced for each training vector. C is a penalty parameter on the training error. The SVM procedure aims at solving the following optimization problem:

$$\underset{\mathbf{w}, b, \xi}{\text{Min}} \qquad \frac{1}{2}\left\|\mathbf{w}\right\|^2 + C\sum_{i=1}^{L}\xi_i \tag{2}$$

subject to

$$\begin{aligned} & y_i^l \cdot (\mathbf{w}^T \cdot \mathbf{x}_i^l + b) \geq 1 - \xi_i && i = 1, ..., L, \\ & \xi_i \geq 0 && i = 1, ..., L. \end{aligned}$$

Notice that the examples that are farthest to the classifier have a higher confidence and are more likely to belong to their corresponding class. The self-training algorithm for reject inference follows:

Algorithm 1 Self-Training for Reject Inference

```
while($\mathscr{U} \neq \emptyset$) do
    train a SVM classifier $f$ using (1) with all data in $\mathscr{L}$;
    use $f$ to classify all unlabeled examples in $\mathscr{U}$;
    transform $f(\mathbf{x_i^u})$ to a probabilistic outcome using Platt's logit link function:

        $P(y = 1|f(\mathbf{x_i^u})) = \frac{\mathbf{1}}{\mathbf{1 + \exp(Af(x_i^u) + B)}}$    (3)

    adjust the cut-off threshold by incorporating a higher risk in unlabeled data:

        $f_\lambda(\mathbf{x_u}) = \mathbf{P(y = 1|f(x_i^u)) - 0.5} + \lambda$    (4)

    select $\mathbf{x}^* \in \mathscr{U}$ with higher confidence $|f_\lambda(x^*)|$;
    $\mathscr{L}$.add(($\mathbf{x}^*, \mathbf{sign}(\mathbf{f}_\lambda(\mathbf{x}^*))$));
    $\mathscr{U}$.remove($\mathbf{x}^*$);
end while;
```

The parameters A and B in the link function (2) are fit using maximum likelihood estimation from the training set [23]. Parameter $\lambda \in [-0.5, 0.5]$ represents the relative risk of the rejected applicants in terms of the accepted examples. $\lambda = 0$ performs standard self training, while a value of λ equals to one of the bounds will assume that all rejected examples belong to an unique class. For example, being the class $+1$ the defaulted loans, $P(y = 1|f)$ means the probability of default for each rejected loan. A negative value of λ considers a higher risk in the rejected loans, and allow rejected examples with a probability of default smaller but close to 0,5 to belong to the positive class (defaulters).

We suggest the following procedure to estimate λ: we define l^- (l^+) as the number of negative (positive) examples in the labeled training data and u^- (u^+) as the number of negative (positive) examples in the unlabeled training data, which at this point we assume known ($l^- + l^+ = l$, $u^- + u^+ = u$). We consider $p_l = \frac{l^-}{l}$ ($p_u = \frac{u^-}{u}$) the probability of belonging to the negative class in the labeled (unlabeled) data, which in credit scoring means the probability of being good customer. We propose $\lambda = p_u - p_l$, which is negative when there is a higher proportion of good customers in the labeled class.

The parameter λ assumes known to probability of being good customer for the rejected loans, which is unknown but can be estimate by understanding the process of credit assessment, and at the end represents a strategic decision. In our experiments the real proportion of good customers in the unlabeled data will be known and we will use this information to obtain λ.

5 Experimental Results

The proposed approach has been applied for two toy data sets, one real-world credit data set from the UCI data repository [16] and one data set from a Chilean financial institution [20]. Next, we describe briefly these data sets and provide the classification results using different reject inference methods.

5.1 Experiments with toy data sets

A Two-dimensional data set have been constructed considering 3 subsets: 100 examples in a training subset with 80 negative instances (good accepted loans, red squares in Figure 1) and 20 positive instances (bad accepted loans, blue diamonds in Figure 1); 100 examples in a second training subset which we consider unlabeled in order to emulate a rejected subset (green circles in Figure 1), with 60 negative instances (good rejected loans) and 40 positive instances (bad rejected loans). Finally we consider a test subset with 100 examples (70 negative instances and 30 positive). The training data set (labeled and unlabeled) and the test data set are drawn from the same distribution. Both variables are generated in order to be useful for the classification task. Figure 1 represents a plot of the training subset of this toy data set.

A second toy data set is obtained using the same data: Considering the same 200 examples for training and the remaining 100 examples for test, we split the training data considering one simple rule: 100 examples above a threshold using one of the variables are consider unlabeled. In this case we do not have overlapped labeled/unlabeled data sets and we have also a higher risk (12 positive instances in the labeled-training subset and 48 positive instances in the unlabeled-training set). Figure 2 represents a plotted view of the training subset of the second toy data set.

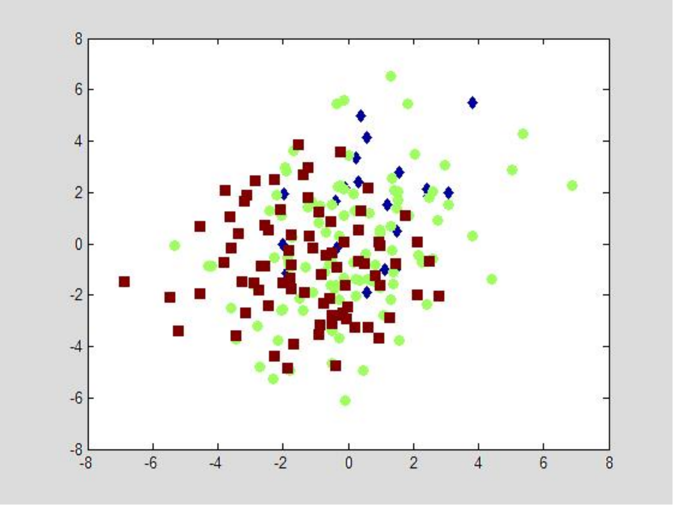

Figure 1: Plot of the data set ”Toy 1”

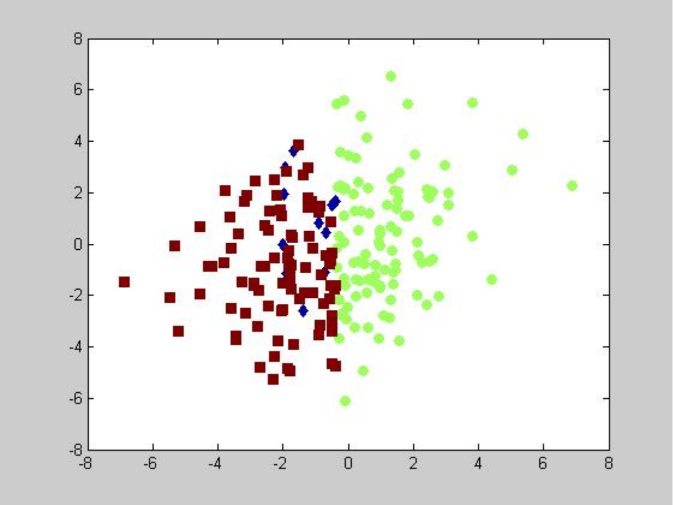

Figure 2: Plot of the data set ”Toy 2”

We test our approach in both data sets together with following solutions for reject inference:

- SVM using only the labeled data set and omitting the unlabeled data set (SVM).
- SVM considering all rejected loans as good loans (SVM+g).

- SVM considering all rejected loans as bad loans (SVM+b).
- SVM using standard self training (SVM+st), which is equivalent to $\lambda = 0$.
- SVM using the proposed modification for self training (λ-SVM).
- SVM using Transductive SVM assuming same proportion of classes in the labeled and unlabeled subset (TSVM).
- SVM using Transductive SVM considering the real proportion of classes in the unlabeled subset (TSVM+p).

Table 1 shows the classification performance (percentage of correctly classified examples over total examples) in the test subset for both data sets:

Table 1: Classification accuracy for two "toy" data sets

	Test Toy 1	Test Toy 2
SVM	79%	70%
SVM+g	70%	70%
SVM+b	59%	72%
SVM+st	80%	70%
λ-SVM	**82**%	**76**%
TSVM	80%	75%
TSVM+p	80%	**76**%

From these experiments we observe a better classification performance with our approach. Transductive SVM performs as good as our approach in the second toy subset but it is much more time-consuming (about 3 hours versus a few seconds). We can also speed up our method to 10 iterations by incorporating 10 examples ($\frac{U}{10}$) to the labeled data set at each iteration, achieving the same classification performance.

From these results we observe that the classification performance on the training set using self training is much better than other method, which can be tricky since the method assumes that the function used to label the rejected examples is classifying all data correctly. It is important to analyze the results only in the test subset and avoid biased conclusions. Self training without any inference of the unlabeled data just reinforce the classifier obtained with the labeled data and does not improve the classification.

5.2 Experiments with a real-world benchmark data set

In order to validate our results with toy examples, we consider the real-world data set German Credit, which consists in 800 examples in a training subset and 200 examples in a test subset, both with approximately 70% of good loans. We split the training subset into a training-labeled subset and a training-unlabeled subset using stratified sampling and selecting approximately 600 instances for the labeled subset and the remaining 200 instances for the unlabeled subset.

We perform 4 different splits of the training subset in order to consider different levels of risk. Table 2 contains the information of the 4 different training subsets:

Table 2: Good loans proportion for German Credit Data

	Gcredit1	Gcredit2	Gcredit3	Gcredit4
good loans labeled subset	70,1%	69,5%	67,6%	66,8%
good loans unlabeled subset	70,1%	70,3%	70,9%	71,2%
total good loans for training	70,1%	70,1%	70,1%	70,1%

We run the mentioned approaches for all 4 training subset. Table 3 shows the classification performance in the test subset for both data sets:

Table 3: Classification accuracy for German Credit Data

	Gcredit1	Gcredit2	Gcredit3	Gcredit4
SVM	78%	76%	76%	75%
SVM+g	70%	70%	73%	70%
SVM+b	58%	66%	67%	65%
SVM+st	78%	76%	77%	75%
λ-SVM	78%	77%	78%	77%
TSVM	77%	74%	76%	71%
TSVM+g	77%	74%	76%	69%

Again our approach outperforms other methods in all 4 training subsets. Notice that our method behaves better in comparison to others in the training sets with a higher difference in terms of risk. When the risk is similar all methods have similar classification performance.

5.3 Experiments with the INDAP data set

The INDAP data set stems from a credit scoring project performed for this Chilean organization. INDAP is the main service provided by the Chilean government that aims at supporting small agricultural enterprises; see `www.indap.cl`. It was founded in 1962 and has more than 100 offices all over Chile serving its more than 100,000 customers [20].

After a feature selection step using the wrapper method HO-SVM [20], the data set is based on 21 variables describing 1,100 observations (767 good and 333 bad customers). We split the whole data set in a training data set with 770 examples and a test data set with 330 examples (both with approximately 70% good customers) using stratified sampling. From the training data set we obtain two subsets: one labeled-training subset emulating accepted loans with 539 observations and 71.6% good customers and one unlabeled-training subset with 231 examples and 65.4% good customers, which represent a sample of rejected loans and will be used for reject inference. Table 4 shows the results

Table 4: Classification accuracy for INDAP data set

	INDAP
SVM	75%
SVM+g	70%
SVM+b	70%
SVM+st	75%
λ-SVM	**76**%
TSVM	**76**%
TSVM+g	75%

in term of accuracy for this data set, considering the experiments mentioned above.

For this data set the proposed approach and TSVM perform slightly better than SVM and standard self training, while traditional reject inference affects negatively in the performance of the classifier.

6 Conclusions

We presented a novel semi-supervised learning approach for reject inference in Credit Scoring using SVM. The intuition behind this method is that we can correct the sample bias by labeling the rejected loans using self learning. Although traditional self learning focus on the unlabeled examples with higher confidence, the important examples in our approach are the less confident ones (rejected loans which the classifier can not certainly conclude if they would have been "bad" or "good" loans): these examples are more likely to be consider "bad" in the adequate proportion in order to adjust the classifier to the real good/bad proportion.

A comparison with other semi-supervised techniques and reject inference strategies for Credit Scoring shows the advantages of our approach:

- It outperforms other reject inference strategies for classification, based on its ability to reproduce the expected risk of the real credit scoring problem ("through the door" population).

- Unlike TSVM, this approach represents an iterative algorithm based on standard SVM, avoiding a complex non-linear optimization problem and ensuring a global optimum solution.

- It can be used with any suitable Kernel function, allowing non-linear classifiers.

- It can be easily generalized to other classification methods, such as logistic regression.

The experiments performed shows that the strategy of considering all rejected loan as ”bad” or ”good” proposed in [26] can affect negatively the accuracy of the classification, and it should be use only in very special cases. On the other hand, reject inference based on data mining strategies, such as self training and transductive algorithms, can improve the classification task by adjusting the expected risk of the unbiased sample of loans, and the significance of this improving is given by the consistency of the current credit system: if accepted and rejected loans differs significatively in terms of good/bad proportion, reject inference using the proposed method helps to obtain a better solution incorporating all available data.

Our algorithm relies on an iterative optimization problem, which is computationally treatable but expensive if the number of input features is large. We could improve its performance by applying filter methods for feature selection before running the algorithm [20]. This way we can identify and remove irrelevant features at low cost. In several Credit Scoring projects we have performed for Chilean financial institutions we used univariate analysis (Chi-Square Test for categorical features and the Kolmogorov-Smirnov Test for continuous ones) as a first filter for features selection with excellent results [20].

Future work has to be done in various directions. First, it would be interesting to improve the proposed technique by incorporating the information of the rules that generated the current score model in order to improve the inference. It would be possible to adjust the original score model by moving the rules according to the performance of the classification and incorporating reject inference based on semi supervised learning. If the original model is not built on a basis of a set of rules, we can extract them using different rule extraction techniques for classification methods [21]. Also interesting is the application of this approach in the domain of spam filtering, where many semi-supervised approaches have been developed in order to improve the classification performance, considering that labeled cases are previously defined by spam filters.

References

[1] Agrawala, A. K. (1970): Learning with a probabilistic teacher. IEEE Transactions on Information Theory, 16:373-379.

[2] Balcan M.F., Blum A., Yang K. (2005): Co-training and expansion: Towards bridging theory and practice. In Saul L.K., Weiss Y., Bottou L.(Eds.), Advances in neural information processing systems 17. Cambridge, MA: MIT Press.

[3] Berger, A. N., Frame, W. S., Miller, N. H. (2005): Credit scoring and the availability, price, and risk of small business credit, Journal of Money, Credit and Banking, 37 (2), 191-222.

[4] Blum, Mitchell, T. (1998): Combining labeled and unlabeled data with co-training. In Proceedings of the 11th Annual Conference on Computational Learning Theory, 92-100.

[5] Castelli, V., Cover, T.M. (1995): On the exponential value of labeled samples. Pattern Recognition Letters, 16:105-111.

[6] Chapelle, O., Zien, A.(2005): Semi-supervised classification by low density separation. Proceeding of the Tenth International Workshop on Artificial Intelligence and Statistic (AISTAT 2005).

[7] Chapelle, Scholkopf, Zien (2005): Semi-Supervised Learning. MIT Press, Cambridge, Massachusetts, London, England.

[8] Chawla, N.V., Karakoulas, G. (2005): Learning from labeled and unlabeled data: An empirical study across techniques and domains. Journal of Artificial Intellegence Research, 23,331-366.

[9] Chen, G., Astebro, T. (2006): A Maximum Likelihood Approach for Reject Inference in Credit Scoring. Rotman School of Management Working Paper No. 07-05.

[10] Chye, K. H., Chin,T. W., Peng G. C. (2004): Credit scoring using data mining techniques. Singapore Management Review. Volume: 26 Issue: 2: 25(23).

[11] Collobert, R., Weston, J., Bottou, L. (2006): Trading convexity for scalability. ICML06, 23rd International Conference on Machine Learning. Pittsburgh, USA.

[12] Culp, M., Michailidis, G. (2007): An iterative algorithm for extending learners to a semisupervised setting. The 2007 Joint Statistical Meetings (JSM).

[13] Goldman, S., Zhou, Y. (2000): Enhancing supervised learning with unlabeled data. Proc. 17th International Conf. on Machine Learning 327-334. Morgan Kaufmann, San Francisco, CA.

[14] Haffari, G., Sarkar, A. (2007): Analysis of semi-supervised learnin g with the Yarowsky algorithm. 23rd Conference on Uncertainty in Artificial Intelligence (UAI).

[15] Hartley, H.O., Rao, J.N.K. (1968): Classification and estimation in analysis of variance problems. Review of International Statistical Institute, 36:141-147.

[16] Hettich, S., Bay, S. D. (1999): The UCI KDD Archive http://kdd.ics.uci.edu. Irvine, CA: University of California, Department of Information and Computer Science.

[17] Joachims, T. (1999): Transductive Inference for Text Classification using Support Vector Machines. International Conference on Machine Learning (ICML).

[18] Johnson, R., Zhang, T. (2007): Two-view feature generation model for semi-supervised learning. The 24th International Conference on Machine Learning.

[19] Maeireizo, B., Litman, D., Hwa, R. (2004): Co-training for predicting emotions with spoken dialogue data. The Companion Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL).

[20] Maldonado, S., Weber, R. (2009): A wrapper method for feature selection using Support Vector Machines. Information Sciences 179 (13), 2208-2217.

[21] Martens, D., Baesens, B., Van Gestel, T., Vanthienen, J. (2006): Comprehensible Credit Scoring Models using Rule Extraction from Support Vector Machines. European Journal of Operational Research.

[22] Nigam, K., Ghani, R. (2000): Analyzing the effectiveness and applicability of co-training. Ninth International Conference on Information and Knowledge Management 86-93.

[23] Platt, J. (1999): Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods, Advances in Large Margin Classifiers. MIT Press, 61-74.

[24] Rosenberg, C., Hebert, M., Schneiderman, H. (2005): Semi-supervised self-training of object detection models. Seventh IEEE Workshop on Applications of Computer Vision.

[25] Scudder, H.J. (1965): Probability of error of some adaptive pattern-recognition machines. IEEE Transactions on Information Theory, 11:363-371.

[26] Siddiqi, N. (2005): Credit Risk Scorecards, Developing and Implementing Intelligent Credit Scoring. Wiley and Sons, First Edition.

[27] Smith, A. (2005): Learning from Data Sets with Missing Labels.

[28] Thomas, L. C. (2002): A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. International Journal of Forecasting, 16(2): 149-162.

[29] Valiant, L.G. (1984): A theory of the learnable. Commun. ACM, 27(11):1134-1142.

[30] Vapnik, V. (1998): Statistical Learning Theory. John Wiley and Sons, New York.

[31] Weston, J., Elisseeff, A., Bakir, G., Sinz, F.: The spider. http://www.kyb.tuebingen.mpg.de/bs/people/spider/.

[32] Xu, J.-M., Fumera, G., Roli, F., Zhou, Z.-H. (2009). Training SpamAssassin with active semi-supervised learning. Proceedings of the 6th Conference on Email and Anti-Spam (CEAS'09), Mountain View, CA.

[33] Yarowsky, D. (1995): Unsupervised word sense disambiguation rivaling supervised methods. Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics 189-196.

[34] Zhou, Y., Goldman, S. (2004): Democratic co-learing.Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI 2004).

[35] Zhu, X. (2007): Semi-Supervised Learning Literature. Survey, Computer Sciences TR 1530, University of. Wisconsin, Madison.