2009-08-21

Learning Without Default: A Study of One-Class Classification and the Low-Default Portfolio Problem

Kenneth Kennedy
Technological University Dublin, kennedykenneth@gmail.com

Brian Mac Namee
Technological University Dublin, brian.macnamee@tudublin.ie

Sarah Jane Delany
Technological University Dublin, sarahjane.delany@tudublin.ie

Recommended Citation

Kennedy,K,MacNamee,B. & Delany,S. (2009) Learning Without Default: A Study of One-Class Classification and the Low-Default Portfolio Problem. *Artificial Intelligence and Cognitive Science, 20th Irish Conference, AICS 2009,* Dublin, Ireland, 19-21, August. doi:10.1007/978-3-642-17080-5_20

Funder: Abbest

Learning Without Default: A Study of One-Class Classification and the Low-Default Portfolio Problem

Kenneth Kennedy[1], Brian Mac Namee[1], and Sarah Jane Delany[2]

[1] School of Computing,
Dublin Institute of Technology, Dublin, Ireland
[2] Digital Media Centre,
Dublin Institute of Technology, Dublin, Ireland
kenneth.kennedy@dit.ie, brian.macnamee@comp.dit.ie,
sarahjane.delany@dmc.dit.ie

Abstract. This paper asks at what level of class imbalance *one-class classifiers* outperform *two-class classifiers* in credit scoring problems in which class imbalance, referred to as the *low-default portfolio problem*, is a serious issue. The question is answered by comparing the performance of a variety of one-class and two-class classifiers on a selection of credit scoring datasets as the class imbalance is manipulated. We also include *random oversampling* as this is one of the most common approaches to addressing class imbalance. This study analyses the suitability and performance of recognised two-class classifiers and one-class classifiers. Based on our study we conclude that the performance of the two-class classifiers deteriorates proportionally to the level of class imbalance. The two-class classifiers outperform one-class classifiers with class imbalance levels down as far as 15% (i.e. the imbalance ratio of minority class to majority class is 15:85). The one-class classifiers, whose performance remains unvaried throughout, are preferred when the minority class constitutes approximately 2% or less of the data. Between an imbalance of 2% to 15% the results are not as conclusive. These results show that one-class classifiers could potentially be used as a solution to the low-default portfolio problem experienced in the credit scoring domain.

1 Introduction

Financial institutions use quantitative *credit scoring* models to assist in the decision of whether or not to grant credit to a credit applicant. The term "credit scoring" is used to describe the process of determining the likelihood that applicants will default on their loan repayments [1]. The outcome of this process results in assigning credit applicants into one of two classes: *accept* (likely to repay or positive) and *reject* (likely to default or negative). Predictive variables extracted from application forms, external data suppliers and existing own-company records allow credit scoring models to yield an estimate of *probability of default* [2]. This decision to accept or reject an applicant for credit is taken by comparing the estimated probability of default with a suitable threshold [2]. Credit scoring models can be divided into two types: (i) *Application scoring* - credit scoring which deals with new applicants and; (ii) *Behavioural scoring* - credit scoring

based on managing existing accounts. This study is confined to application scoring. Credit scoring is used inter-changeably with the term application scoring throughout.

A particular difficulty with building credit scoring models is that the data used to build models is historical data detailing the performance of customers granted credit in the past (i.e. did they or did they not default?). However, the vast majority of customers do not default and so the number of defaulters represented in training sets is typically very low. Furthermore, when defaults do occur they tend to be cyclical, for example a recession can result in a cluster of defaults occurring. This leads to the *low-default portfolio problem* and means that credit scoring datasets are usually heavily imbalanced. [3] report that among the seven largest UK banks, 32% of retail exposures secured by residential properties will suffer from insufficient default data to give a satistically significant estimate. According to the Council of Mortgage Lenders (CML)[3], in the UK for the second quarter of 2009 there were 11,400 cases of possession, equivalent to one mortgage in 1,000. Apart from the academic challenges that arise from the low-default portfolio problem, it is also of considerable practical importance. Even a small improvement of a fraction of a percent in the accuracy of credit scoring might translate into significant future saving [1, 4].

Previously [5] reported that with 5% or lower minority class data, one-class classifiers outperform two-class classifiers. It should be noted that this study used support vector-based classifiers only and the performance of the one-class classifiers on real world datasets was optimised using training data from both classes. A similiar study by the same authors [6] found that one-class classifiers trained on one-class only are prefered with 1% or lower minority class data. [7] use two high dimensional real world datasets and reported that with approximately 3% or lower minority class data, the performance of one-class support vector machine (SVM) [8] surpassed that of the two-class SVM [9].

In this paper we will compare *one-class classification* (OCC) methods with more common two-class classification approaches on three credit scoring datasets over a range of class imbalance ratios. The purpose of this study is to determine at what level of class imbalance the performance of OCC methods outrank the performance of two-class approaches. To the best of our knowledge, no attempt has been made to examine one-class classifiers as a solution to the low-default portfolio problem. The remainder of this paper is organised as follows: a short overview of credit scoring is given in Section 2, followed by a discussion of classification techniques in Section 3. Section 4 describes the classification performance criteria of the experiments and then evaluates classifier performance. Section 5 presents conclusions and future work.

2 Credit Scoring

The recent subprime mortgage crisis in the USA has caused some companies the loss of billions of dollars due to customers' defaults. Effective credit risk assessment is now

[3] The Council of Mortgage Lenders is an industry body whose members are banks, building societies and other lenders who together undertake around 98% of all residential mortgage lending in the UK. There are 11.1 million mortgages in the UK, with loans worth over £1.2 trillion.

recognised as a crucial factor to gaining a competitive advantage which can help financial institutions to grant credit to creditworthy customers and reject non-creditworthy customers. According to the CML, UK gross mortgage lending for the second quarter of 2009 was estimated to total £33,902 million. It is therefore legitimate to conclude that a small improvement in the accuracy of credit scoring has positive financial consequences. Another practical consideration is Basel II regulation [10]. Under this accord, using the internal ratings based (IRB) approach, financial institutions calculate their own risk parameters (e.g. probability of default) in order to calculate risk weighted assets. The risk weighted assets help determine the minimum capital requirements that the banks are required to retain, and act as a buffer against unexpected losses. Using the IRB approach, financial institutions can create credit scoring models more customised to certain risk sensitivities. Such legislation serves to increase the importance of credit scoring whilst creating new challenges.

Many classification techniques have been used for credit scoring [11], some of which include traditional statistical methods such as logistic regression; non-parametric statistical methods, such as *k*-nearest neighbour; and sophisticated methods such as neural networks.

3 Classification Techniques

This section lists the classifiers used in our study. The following two-class classifiers were assessed: (i) Logistic regression[4]; (ii) Naïve Bayes [13][4]; (iii) Artificial neural network using a multilayer perceptron (MLP)[5]; and (iv) Support Vector Machines [9] as they have been shown to perform well when applied to credit scoring problems in the past [11]. For all of the two-class classifiers a cut-off score is applied to the classifier output score, data instances above the cut-off are assigned to the positive class and those with scores below to the negative class.

3.1 One Class Classification

One-class classifiers are constructed to recognise a target class from all other classes. Other synonymous terms used in the literature also include: *outlier detection* [14], *novelty detection* [15], *concept learning* [16] and *data description* [17]. One-class classifiers can be categorised into three types:(i) *density-based*; (ii) *boundary-based* and; (iii) *reconstruction-based*. In all three types, two distinct elements can be identified. The first element is a measure for the distance $d(z)$ or resemblance $p(z)$ of an object z to the target class. The second element is a user-defined threshold, θ, on this distance or resemblance. New objects are accepted when the distance to the target class is less than the value of θ or when the resemblance is greater than the value of θ. OCC methods differ in their definition of $p(z)$ or $d(z)$, and in their optimisation of thresholds with respect to the training set [18]. A comprehensive review of OCC methods and techniques is available in [19, 20]. In the current study we select five common OCC techniques: Gaussian and

[4] See [12] for further details on the use of this technique in credit scoring
[5] See [4] for further details on the use of artificial neural networks in credit scoring

Naïve Parzen (density-based types), Support Vector Data Description (SVDD) and k-Nearest Neighbour (k-NN) (boundary-based types), and k-means (reconstruction-based type).

Gaussian Model [18]: This method assumes that the data is distributed according to the normal (Gaussian) distribution. The mean and covariance matrix is estimated from the data, and instances located in the two tails are considered outliers. A user-defined parameter r can be used to add regularisation to the estimated covariance matrix.

Naïve Parzen [18]: This technique is a simplification of the Parzen density estimator inspired by the Naïve Bayes approach [21]. A Parzen density is estimated for each separate feature dimension, and the probabilities are multiplied to give the final target probability [21].

Support Vector Data Description [18, 22]: The SVDD separates the data of interest from different classes by placing a hypersphere around the class of objects that are represented by the training set from all other possible objects in the object space. The hypersphere is defined by a centre a and a radius R. The aforementioned threshold, θ, can be supplied to allow the hypersphere model of the SVDD to reject a fraction of the training objects, which sufficiently decreases the volume of the hypersphere. The boundaries of the hypersphere can be made more flexible by introducing kernel functions of user-defined width.

***k*-NN** [18]: k-NN finds the distance of a test object x to its k-th nearest neighbour in the training set, the distance of this nearest neighbour and its k-th nearest neighbour in the training set is also found. Based on the quotient between the first and second distance and an appropriate threshold value, x may either be rejected as being an outlier, or accepted as being part of the target class.

***k*-means** [18]: k-means clustering is one of the simplest reconstruction methods. In order to perform k-means clustering for OCC, it is assumed that the data is clustered and can be described by a set of prototype vectors. To classify a new object, its distance to all the prototypes is measured and averaged. This is used to score the extent to which it is an outlier.

3.2 Evaluation Experiment

The aim of the evaluation is to compare the performance of one-class classifiers with two-class classifiers and assess whether one-class classifiers can successfully identify defaulters, and at what level of class imbalance their performance is superior to that of the two-class classifiers. This section describes the datasets, and the evaluation measures and methodology. Finally, experimental results are presented and discussed.

3.3 Datasets

Three real-world datasets are used in our experiments: the Australian[6], Japanese[7] and German[8] credit datasets, all of which are available from the UCI Repository of Machine Learning Databases [23]. Table 1 describes the characteristics of the datasets.

[6] http://archive.ics.uci.edu/ml/datasets/Statlog+(Australian+Credit+Approval)
[7] http://mlr.cs.umass.edu/ml/datasets/Japanese+Credit+Screening
[8] http://archive.ics.uci.edu/ml/datasets/Statlog+(German+Credit+Data)

The class ratio of accept instances to reject instances is included. The Australian credit dataset consists of 307 instances of creditworthy applicants and 383 instances of non-creditworthy applicants. The Japanese dataset describes credit card application approval. After deleting the data with missing attribute values, there are 653 instances, with 357 instances granted credit and 296 instances refused credit. The German credit scoring data is imbalanced to a greater extent, consisting of 700 creditworthy applicants and 300 non-creditworthy applicants. In all cases the variables used describe important features of a customer such as credit history, personal information and details of the credit requested. The numeric features of all three datasets are normalised.

Table 1. Characteristics of the datasets used in the experimentation

Dataset	# Classes	Accept:Reject	# Nominal features	# Numeric features	# Boolean features
Australian	2	45:55	6	8	0
Japanese	2	55:45	6	6	3
German	2	70:30	7	13	0

3.4 Assessment Measures

To assess the classification results we count the number of true positive (TP), true negative (TN), false positive (FP) (classified as positive, but actually negative) and false negative (FN) (classified as negative, but actually positive) examples in a given test set. We use *Sensitivity*, *Specificity*, as used by [11], and the *harmonic mean* of both of these scores to measure the classification quality of all classifiers used in our study. Sensitivity is calculated as: $\frac{TP}{TP+FN}$ and measures the proportion of positive (accept) examples that are predicted to be positive. Specificity, calculated as: $\frac{TN}{TN+FP}$, measures the proportion of negative (reject) examples that are predicted to be negative. As per [24], in order to provide a suitable composite measure of sensitivity and specificity we employ the harmonic mean, which corresponds to a particular adaptation of the F-measure [25].

$$Harmonic\ Mean = \frac{2 * Sensitivity * Specificity}{Sensitivity + Specificity} \tag{1}$$

3.5 Experimental Procedure

Each dataset was divided into training, validation and testing data by stratified random sampling, in which there were 55% training, 15% validation and 30% testing examples per dataset. The process of training, validation and testing was conducted 10 times, the average results are reported. Initially for all three datasets the two-class classifiers were trained on the training data, the validation data was used to tune the model and eventually their performance was assessed on the test data. The sensitivity, specificity and harmonic mean were recorded. Then for all three datasets, the number of negative instances in the training dataset was randomly reduced by 10%. It is necessary to balance

the datasets in order to avoid disproportionate outputs to the majority class. To achieve this random oversampling was performed on the remaining data instances of the negative class. The validation and test sets remained unchanged throughout this process. The classifiers were retrained and reassessed on the test dataset. This process was repeated until the number of negative examples in the training set reached zero.

While simplistic, random oversampling has performed well in empirical studies (e.g. [26]) even when compared to other, more complicated oversampling methods [27]. As oversampling only replicates existing data instances, it can be argued that it does not add any actual data to the dataset [5, 27]. Figure 1 illustrates the effect of not oversampling the minority class on the Australian dataset. Without balancing the training set, the performance of the two-class classifiers (particularly the SVMs) deteriorates. The Naïve Bayes classifier trained on the Australian and Japanese dataset proved to be an exception. Oversampling actually weakened the performance of the Naïve Bayes classifier to a small degree, as illustrated in Figure 1. After training, there is a possibility that the Naïve Bayes classifier fits the data well, but performs poorly at predicting new values. For example, it may be strongly affected by extreme attribute values and other artifacts of the training dataset.

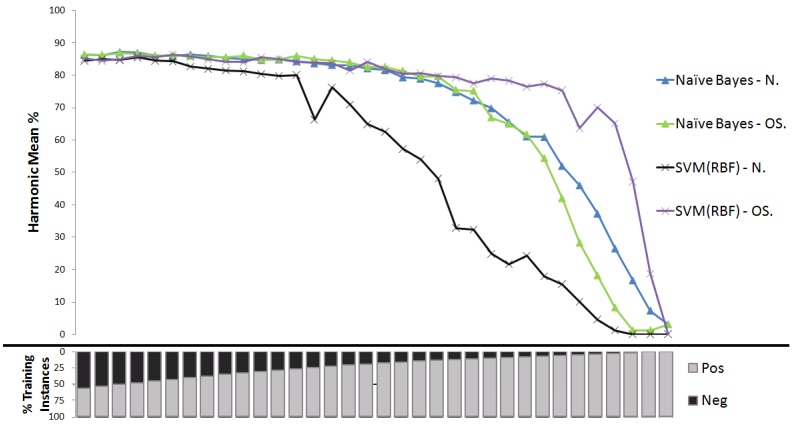

Fig. 1. Australian Harmonic Mean: % of available Training Instances. A comparison of oversampling (OS) and no oversampling (N).

The Data Description toolbox[9] is an open source Matlab library of one-class classifiers and was used to implement the OCC techniques used in this study. When training the one-class classifiers only data from one class was employed, the positive class (creditworthy applicants). The validation data (consisting of two-classes) was used to tune the models and their performance was assessed on the test data. For the one-class k-means classifier the number of clusters was fixed at 10. For the one-class k-NN classifier the number of neighbours was set to 10. This figure was selected in keeping with [11],

[9] (http://ict.ewi.tudelft.nl/~davidt/dd_tools.html)

who used *10*-NN two-class classifier. The one-class Naïve Parzen classifier required no parameter tuning. The SVDD used the Gaussian kernel (default setting).

All the two-class classifiers were implemented using Weka [28] release 3.6.0. The Naïve Bayes classifier used a supervised discretisation algorithm to convert numeric attributes to nominal attributes. The logistic regression classifier was optimised for the ridge value in the log-likelihood. The neural net was implemented using a multi-layer perceptron (MLP). The number of hidden neurons was defined as (*#attributes* + *classes*)/2. The MLP was optimised using the rate of learning. The SVM was implemented using Lib-SVM [29] using a radial basis function (RBF) kernel and adopted a grid search mechanism to tune the width γ of the RBF kernel and the cost parameter C. Results for a SVM using a linear kernel are also included.

4 Results and discussion

Table 2 displays the performance of the the two-class classifiers across the datasets. The classifiers have been trained on data containing: all of the available defaulters (100%); one-fifth of the available defaulters (20%) and so on until none of the defaulters are used (0%). The deterioration in the two-class classifiers is largely due to their inability to correctly identify the increasingly rare defaulters. Eventually, in almost all cases, their sensitivity rate hits 100% because in the absence of defaulters they identify all test set instances as non-defaulters. Of the two-class classifiers, based on the harmonic

Table 2. Sensitivity, specificity and harmonic mean (HM) using the Logistic Regression (LR), Naïve Bayes (NB), MLP, SVM RBF kernel (SVM-R), and SVM Linear kernel (SVM-L) classifiers. Best performing HM for each dataset is underlined.

		100%			20%			10%			5%			0%		
Dataset	Classifier	Sens	Spec	HM	Sens	Spec	HM	Sens	Spec	HM	Sens	Spec	HM	Sens	Spec	HM
Aus	LR	83.88	88.21	85.93	84.05	82.70	83.12	87.25	75.76	80.91	86.57	66.48	75.02	100	0	0
	NB	81.28	92.26	86.38	84.96	83.34	84.00	89.73	65.03	75.20	95.47	27.57	42.02	99.38	1.59	3.09
	MLP	81.10	89.61	85.05	90.41	83.43	86.62	90.71	79.15	84.25	90.18	67.64	75.97	100	0	0
	SVM-R	83.27	87.13	85.04	85.84	77.99	81.45	83.47	72.92	77.44	85.58	68.22	75.33	100	0	0
	SVM-L	91.27	85.18	88.04	90.28	85.66	87.78	88.48	81.47	84.62	89.06	70.94	78.07	100	0	0
Jap	LR	84.99	82.72	83.69	88.26	74.08	80.37	89.42	69.77	78.28	90.57	65.02	75.38	100	0	0
	NB	89.49	79.75	84.23	91.58	71.54	80.16	95.68	44.14	60.13	99.71	6.00	10.99	100	0.70	1.37
	MLP	85.26	87.25	86.07	89.17	75.78	81.57	92.09	61.95	72.84	97.24	37.46	52.96	100	0	0
	SVM-R	84.42	87.95	85.99	87.59	83.38	85.13	88.31	72.10	78.96	88.42	63.10	72.90	100	0	0
	SVM-L	81.63	93.23	86.99	85.76	83.74	84.45	88.38	74.49	80.27	90.32	58.57	70.17	100	0	0
Ger	LR	72.98	69.18	70.93	78.39	53.99	63.59	81.54	47.07	59.31	88.39	27.59	41.29	100	0	0
	NB	75.12	66.19	70.22	85.38	37.16	50.50	96.09	9.61	17.18	99.07	2.13	4.04	99.95	0.31	0.62
	MLP	75.73	57.75	64.22	84.77	34.52	46.62	91.10	24.12	37.58	84.70	24.40	22.79	100	0	0
	SVM-R	73.32	68.11	70.43	74.21	56.65	63.83	77.89	44.94	54.36	82.27	30.85	43.02	100	0	0
	SVM-L	72.62	69.46	70.89	71.84	54.17	61.28	76.20	47.01	57.30	85.21	27.54	39.43	100	0	0

mean across the range of all class imbalances, there is very little to distinguish the performance of logistic regression, SVM with RBF kernel and SVM with linear kernel, as exemplified in Table 2. Overall, the SVM linear kernel performs fractionally better than logistic regression followed closely by the SVM with RBF kernel. However it should be noted that this is a generalised logistic regression model and financial institutions typically have at their disposal methods to increase and extend its accuracy and flexibility [30]. The Naïve Bayes performs worst of the two-class classifiers.

Table 3 displays the results of the selected one-class classifiers. Overall, the Gaussian and Naïve Parzen models appear to perform best. Both of these models are density-based methods. This approach works very well when a good probability model is assumed and the sample size is sufficiently large [18]. Excluding the Gaussian one-class

Table 3. Sensitivity, specificity and harmonic mean (HM), along with standard deviation, for the one-class classifiers. Best performing HM for each dataset is underlined.

	Australian			Japanese			German		
Classifier	Sens	Spec	HM	Sens	Spec	HM	Sens	Spec	HM
Gaussian	74.43	82.43	77.77 (2.97)	74.03	76.46	74.77 (3.03)	57.05	59.38	56.76 (3.15)
Naïve Parzen	59.02	71.10	63.52 (3.81)	77.26	74.39	75.48 (1.81)	58.90	50.44	53.08 (4.24)
k-NN(10)	66.53	68.25	65.78 (3.14)	71.33	63.66	66.73 (3.35)	59.38	54.45	56.03 (2.13)
k-means(10)	67.24	66.45	64.56 (3.86)	66.13	64.88	64.23 (2.60)	61.69	53.54	56.54 (2.43)
SVDD	60.85	70.87	65.17 (3.82)	67.48	60.62	61.78 (4.75)	65.72	47.27	53.69 (3.57)

classifier, the performance of the one-class classifiers on the Australian dataset is rather ordinary. There are a number of factors that might contribute to this. Two customers with similar characteristics can easily belong to different classes [31]. Also, credit scoring datasets are typically very noisy [11], particularly the Australian dataset [32].

Figures 2, 3 and 4 display the test set harmonic mean of all 10 classifiers for each of the datasets. The rate of training set transformation, in terms of the number of positive and negative instances, is displayed as a percentage bar beneath each harmonic mean graph. It is evident from Figures 2, 3 and 4 that initially the two-class classifiers outperform the one-class classifiers. However, as the number of defaulters are gradually removed from the training sets the performance of the two-class classifiers begins to deteriorate. As the one-class classifiers are trained using only non-defaulters their performance remains constant throughout.

The crossover in performance between the best one-class classifier and worst two-class classifier on the Australian test set occurs when the training set is approximately 14% minority class data. For the Japanese test set, the crossover between the best one-class classifier and worst two-class classifier occurs when the training set is approximately 11% minority class data. The crossover between the best one-class classifier and worst two-class classifier on the German test set occurs when the training set is approximately 20% minority class data. Based on these figures it would appear that two-class classifiers outperform one-class classifiers with 15% or more of the minority class data.

With the Australian test set, the crossover in performance between the best two-class classifier and the worst one-class classifier occurs when the training set is approximately 2% minority class data. For the Japanese test set, this crossover occurs between the best two-class classifier and the worst one-class classifier occurs when the training set is approximately 2% of minority class data. As the German dataset begins with a positive:negative ratio of 70:30, the harmonic mean of the two-class classifiers declines quickest of all three datasets. The crossover in performance between the best two-class classifier and worst one-class classifier occurs when the training set is approximately 3% of minority class data. These figures indicate that one-class classifiers outperform two-class classifiers with 2% of minority class data. This suggests that two-class classification methods are relatively robust to imbalanced data and that OCC methods should only be considered in the most extreme cases.

With minority class data of between 2% and 15% the distinction between OCC and two-class classification methods is less clear cut. However, the crossover between the best one-class classifier and best two-class classifier on the Australian test set occurs when the training set is approximately 5% minority class data. For the Japanese test set, the crossover between the best one-class classifier and best two-class classifier occurs when the training set is approximately 4% minority class data. The crossover between the best one-class classifier and best two-class classifier on the German test set occurs when the training set is 3% minority class data. Therefore with approximately 4% or less minority class data, one-class classifiers, under certain conditions, can be considered ahead of two-class classifiers.

5 Conclusions

This study asked at what level of class imbalance the performance of OCC techniques outperform two-class classification techniques for credit scoring problems. Class imbalance is a particularly important issue in credit scoring applications due to the low-default portfolio problem. The experiments were conducted using three real-world credit scoring datasets. It was found that, initially, the two-class classifiers outperform the one-class classifiers. However as the the rate of class imbalance increases and the performance of the two-class classifiers falls off.

With 2% or lower of minority (reject or negative) class data, one-class classifiers are more accurate than two-class classifiers. Conversely, with 15% or higher minority class data, two-class classifiers clearly outperform one-class classifiers. With an imbalance between 2% and 15% of minority class data, the results are not as conclusive, however with 4% or lower of minority class data, certain one-class classifiers outperform two-class classifiers. Therefore we can conclude that one-class classifiers offer a viable solution to the low-default portfolio problem when the class imbalance is severe, and so warrant further research as a solution to the low-default portfolio problem.

For the two-class classifiers, the harmonic mean of the sensitivity and specificity was calculated assuming a default cut-off value on the classifier's output. This may, however, not be the most appropriate threshold to use for more skewed datasets as some classifiers have a tendency to always predict the majority class yielding 100%

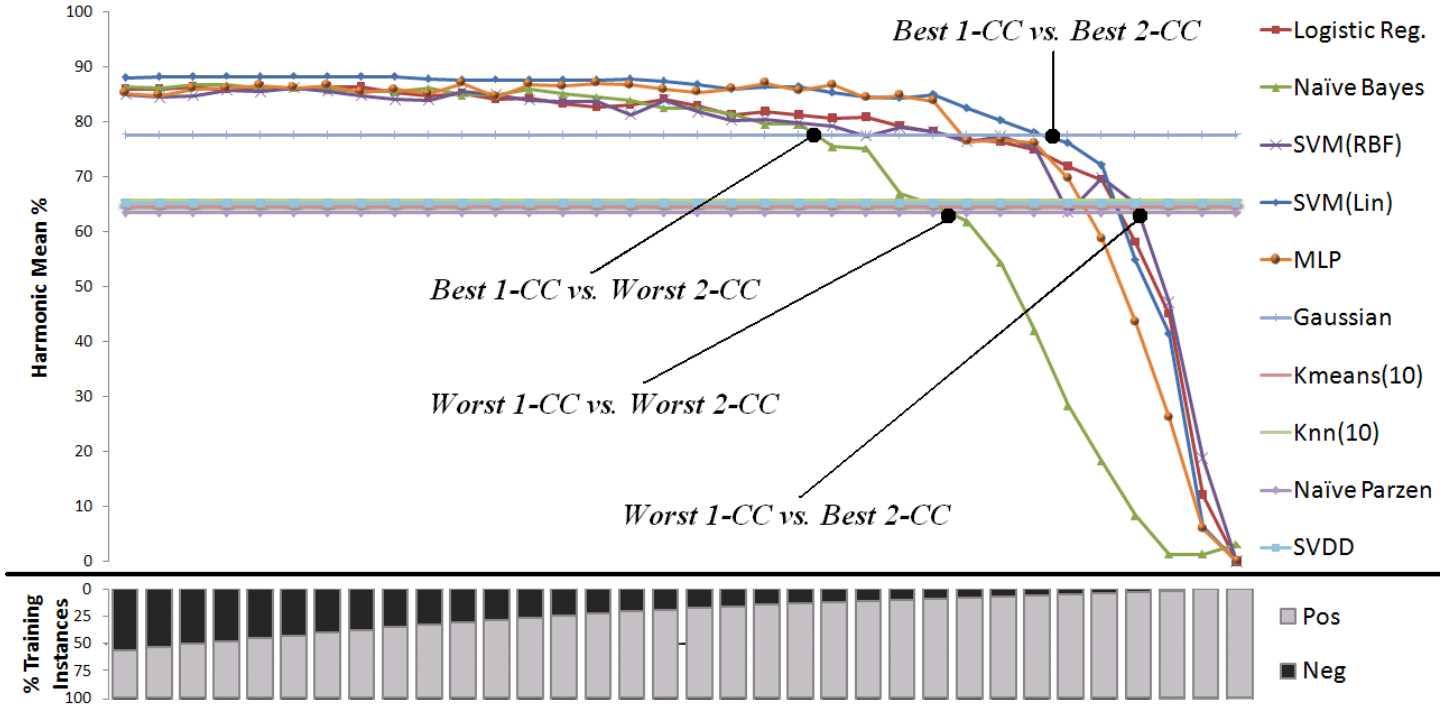

Fig. 2. Australian Harmonic Mean: % of Available Training Instances. Crossover points of best/worst one-class classifier (1-CC) and two-class classifier (2-CC) are also identified.

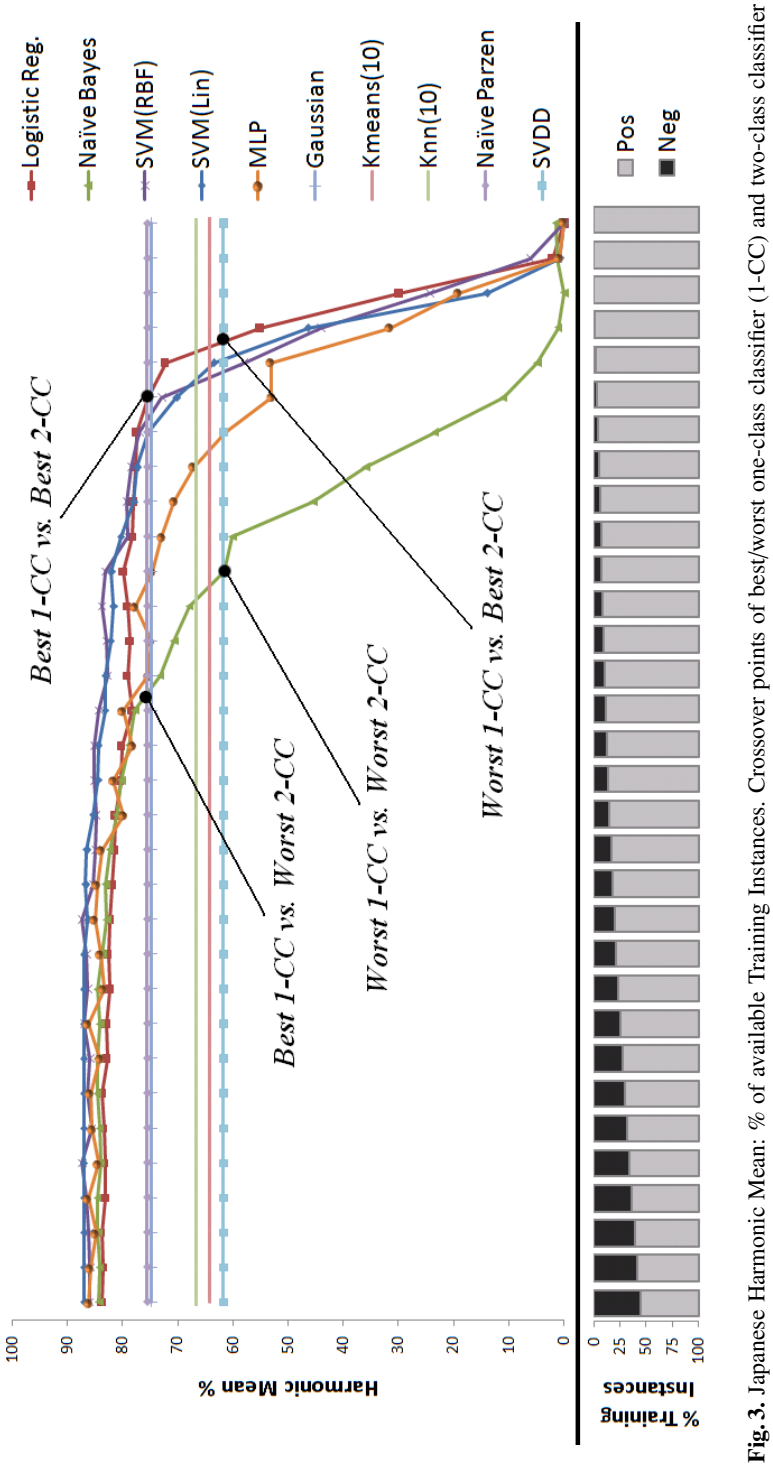

Fig. 3. Japanese Harmonic Mean: % of available Training Instances. Crossover points of best/worst one-class classifier (1-CC) and two-class classifier (2-CC) are also identified.

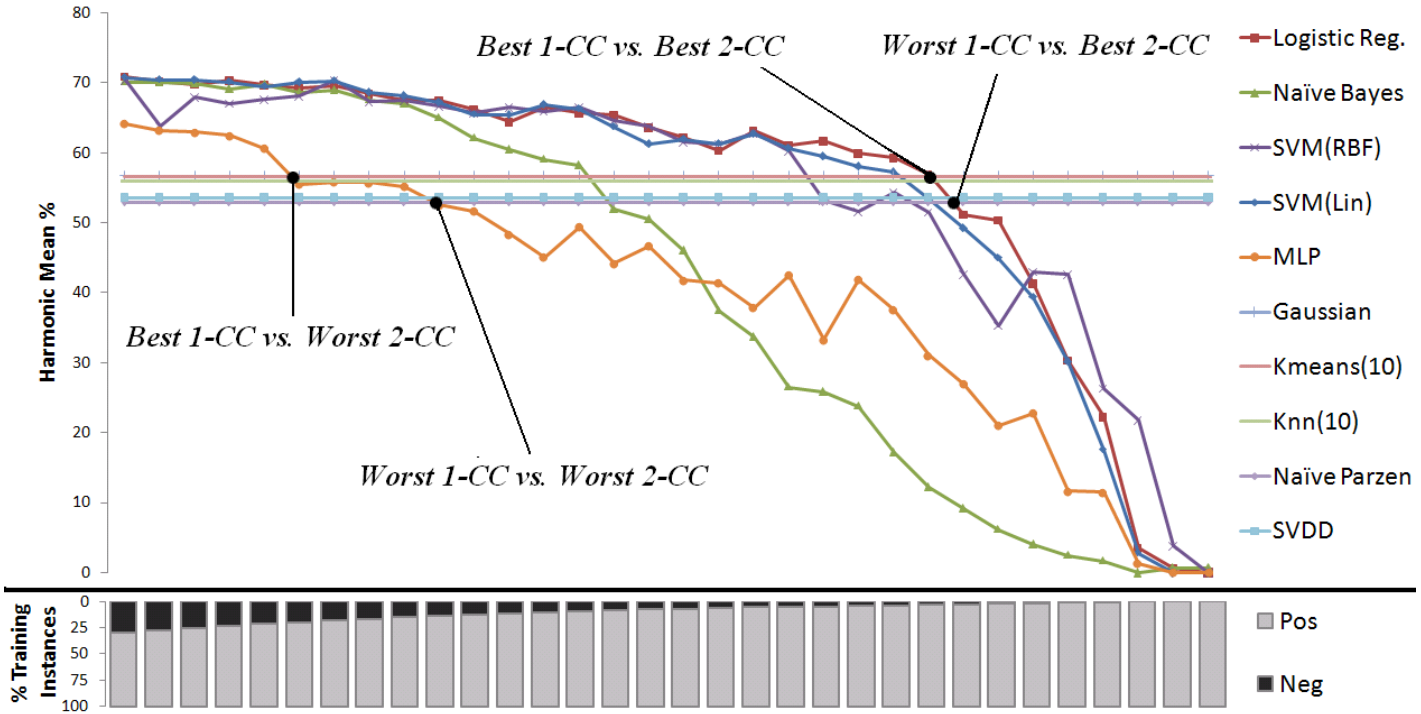

Fig. 4. German Harmonic Mean: % of available Training Instances. Crossover points of best/worst one-class classifier (1-CC) and two-class classifier (2-CC) are also identified.

sensitivity and 0% specificity. However, by applying random oversampling this concern was alleviated to some degree.

In a future experiment we will use a validation set to determine the optimal cut-off threshold for each classifier at each level of class imbalance. A two-dimensional graph called the receiver operating characteristic (ROC) curve is commonly used, particularly with class imbalance, to present the results of two-class classifiers. However, a debate exists on the appropriate application of ROC curves [33]. When a large skew in the class distribution occurs, ROC curves sometimes provide an overly optimistic view of an algorithm's performance [34]. Furthermore, ROC curves can be unreliable in the case of severe class imbalance [35]. Cost curves [36] could also be used and a comparison between the suitability of both measures should be discussed.

Despite the absence of defaulters from the training set, one-class classifiers proved successful at identifying defaulters. Conversely, having been trained exclusively on non-defaulters, one-class classifiers performance at identifying the creditworthy cases was rather unremarkable. This leads to an obvious direction of future research: investigating the performance of classifier ensembles consisting of a combination of several one-class and two-class classifiers whose classification decisions are computed based on various voting schemes.

Finally, there are many other factors that influence the low-default portfolio problem, such as the size of the data, data fragmentation and the complexity of the inputs to name a few [37, 38]. The low-default portfolio problem needs to be analysed with respect to them.

Acknowledgment

The authors would like to thank Pádraig Cunningham (University College Dublin) for his valuable remarks and suggestions.

References

1. Hand, D.J., Henley, W.E.: Statistical classification methods in consumer credit scoring: a review. Journal of the Royal Statistical Society. Series A (1997) 523–541
2. Verstraeten, G., den Poel, D.V.: The impact of sample bias on consumer credit scoring performance and profitability. Journal of the operational research society **56** (2004) 981–992
3. Joint British Bankers Asc, London Investment Banking Asc, I.S., Group, D.A.I.W.: The irb approach for low default portfolios (ldps)- recommendations of the joint bba, liba, isda industry working group. BBA, LIBA, ISDA Working Paper (2004)
4. West, D.: Neural network credit scoring models. Computers and OR **27** (2000) 1131–1152
5. Lee, H., Cho, S.: The novelty detection approach for different degrees of class imbalance. Lecture Notes in Computer Science **4233** (2006) 21
6. Lee, H., Cho, S.: Focusing on non-respondents: Response modeling with novelty detectors. Expert Systems with Applications **33** (2007) 522–530
7. Raskutti, B., Kowalczyk, A.: Extreme re-balancing for SVMs: a case study. ACM SIGKDD Explorations Newsletter **6** (2004) 60–69
8. Scholkopf, B., Platt, J.C., Shawe-Taylor, J., Smola, A.J., Williamson, R.C.: Estimating the support of a high-dimensional distribution. Neural computation **13** (2001) 1443–1471

9. Vapnik, V.: The nature of statistical learning theory. Springer-Verlag. New York (1995)
10. Bank for Intl. Settlements: Basel II: intl. convergence of capital measurement and capital standards: a revised framework. BIS (2004)
11. Baesens, B., Gestel, T.V., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. JORS **54** (2003) 627–635
12. Thomas, L.C., Oliver, R.W., Hand, D.J.: A survey of the issues in consumer credit modelling research. Journal of the Operational Research Society **56** (2005) 1006–1015
13. Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. (1973)
14. Ritter, G., Gallegos, M.T.: Outliers in statistical pattern recognition and an application to automatic chromosome classification. Pattern Recognition Letters **18** (1997) 525–540
15. Bishop, C.M.: Novelty detection and neural network validation. IEE Proceedings-Vision, Image and Signal processing **141** (1994) 217–222
16. Japkowicz, N., Myers, C., Gluck, M.: A novelty detection approach to classification. In: In Proceedings of the Fourteenth Joint Conference on Artificial Intelligence. (1995)
17. Tax, D.M.J., Duin, R.P.W.: Support vector domain description. Pattern Recognition Letters **20** (1999) 1191–1199
18. Tax, D.: One-class classification. Unpub. doc/dis, Delft University of Technology (2001)
19. Hodge, V., Austin, J.: A survey of outlier detection methodologies. Artificial Intelligence Review **22** (2004) 85–126
20. Chandola, V., Banerjee, A., Kumar, V.: Anomaly detection: A survey. ACM (2009)
21. Tax, D.M.J., Duin, R.P.W.: Characterizing one-class datasets. In: Proceedings of the 16th Annual Symposium of the Pattern Recognition Assoc. of S. Africa, Citeseer (2005) 21–26
22. Tax, D.M.J., Duin, R.P.W.: Support vector data description. ML **54** (2004) 45–66
23. Asuncion, A., Newman, D.: UCI machine learning repository. University of California, Irvine, School of Information and Computer Sciences (2007)
24. Hoff, K.J., Tech, M., Lingner, T., Daniel, R., Morgenstern, B., Meinicke, P.: Gene prediction in metagenomic fragments. BMC bioinf. **9** (2008) 217
25. Rijsbergen, C.J.V.: Information Retrieval. Butterworths, London (1979)
26. Batista, G.E., Prati, R.C., Monard, M.C.: A study of the behavior of several methods for balancing machine learning training data. ACM SIGKDD Explrs. Newsl. **6** (2004) 20–29
27. Liu, A., Ghosh, J., Martin, C.: Generative oversampling for mining imbalanced datasets. In: Proceedings of the 2007 International Conference on Data Mining, DMIN. (2007) 25–28
28. Witten, I.H., Frank, E. Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann Publishers (2000) 265–320
29. Chang, C.C., Lin, C.J.: (LIBSVM: a library for support vector machines)
30. Ong, C.S., Huang, J.J., Tzeng, G.H.: Building credit scoring models using genetic programming. Expert Systems with Applications **29** (2005) 41–47
31. Hand, D.J.: Consumer credit and statistics. Statistics in Finance (1998) 69–81
32. Quinlan, J.R.: Simplifying decision trees. Machine Intel **27** (1987) 234
33. Elkan, K.: Invited talk- the real challenges in data mining- a contrarian view. (2003)
34. Davis, J., Goadrich, M.: The relationship between precision-recall and ROC curves. In: Proc of the 23rd intl conf on ML, ACM NY, USA (2006) 233–240
35. Elazmeh, W., Japkowicz, N., Matwin, S.: Evaluating misclassifications in imbalanced data. Lecture Notes in Computer Science **4212** (2006) 126
36. Drummond, C., Holte, R.C.: Explicitly representing expected cost: An alternative to ROC representation. In: Proc of 6th ACM SIGKDD, ACM NY, USA (2000) 198–207
37. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis **6** (2002) 429–449
38. Weiss, G.M.: Mining with rarity: a unifying framework. ACM SIGKDD Explorations Newsletter **6** (2004) 7–19