Clustering Large Credit Client Data Sets for Classification with SVM

Ralf Stecking[1] and Klaus B. Schebesch[2]

[1] Department of Economics, University of Oldenburg, D-26111 Oldenburg, Germany ralf.w.stecking@uni-oldenburg.de
[2] Faculty of Economics, University "Vasile Goldiş", Arad, Romania kbsbase@gmx.de

1 Motivation and Background

Credit client scoring on medium sized credit client data sets can be accomplished very successfully by means of support vector machines (SVM) which are a powerful and robust machine learning method for classification. However, most real life credit client data sets are much bigger, containing several ten or hundred thousand credit client records. Such data sets impose severe computational barriers for SVM, and may even lead to complete intractability in practice. Such intractability is a general and highly relevant problem in other related classification domains like e-commerce. Hence, new effective solutions which avoid direct extensive training on the complete data are in high demand. In order to design such a non-extensive training method for our credit client data we propose to divide the large data set into a much smaller but still sufficient number of homogeneous clusters. These clusters are symbolic objects that will be represented in many alternative ways e.g. by intervals, means, frequency tables or histograms. Classification of such more involved data representations requires an adequate data coding to be combined with appropriate multivariate analysis tools. We show how to construct good and bad credit client clusters that can be classified with SVM. In this way, very large data sets can be made accessible to SVM with non linear kernels and to SVM which handle non standard situations of classification (with unequal class sizes and asymmetric misclassification costs), which are very important in practice. In using SVM we can profit from information inherent to support vectors. We can derive region information from different types of such support vectors, which in turn helps to get more insight into the data clusters, which can be used to compare alternative cluster solutions or to learn a more appropriate distance function for the process of clustering. Finally, we show how to forecast future credit client behaviour with our model, taking into account that a newly arriving credit client record has to be coded in an appropriate way before using the cluster based SVM forecasting model.

The search for improved out-of-sample performance of empirical classification tasks includes various procedures which attempt to guide the use of powerful standard tools like Support Vector Machines (SVM). Given a set of $N > 0$ training examples $\{x_i, y_i\}$, $i = 1, ..., N$, with $x_i \in \mathbb{R}^m$ (m input features) and labels $y_i \in \{-1, 1\}$ (for simplicity, say). Support vectors are those training examples which are near the class boundaries of a SVM solution and which permit training of a classifier with the same out-of-sample performance as a classifier trained on all the other (redundant, etc.) training examples as well. In order to follow the exposition in the sequel it suffices to know that the SVM finally produces a decision rule (a separating function) of the type $y^{pred} = \mathbf{sign}(s(x)) = \mathbf{sign}\Big(\sum_{i=1}^{N} y_i \alpha_i^* k(x_i, x) + b^*\Big)$, with $0 \leq \alpha_i^* \leq C$ and b^* the result of the SVM optimization. Important is to note that $\alpha_i^* > 0$ (a support vector i referred to as "SV" if $\alpha_i^* < C$ and as "BSV" if $\alpha_i^* = C$ in the sequel) actively contribute to the decision function by invoking the ith training example via a user defined kernel $k(,)$ which in most cases is selected to be a semi-positive, symmetric and distance dependent function.

Several comparative studies of credit scoring and rating approaches with SVM have been published. Baesens et al. [1] report on benchmarking SVM with other classifiers using several different data sets. They find SVM performing well, but with traditional methods being competitive, especially when there is weak non-linearity in credit scoring data sets. Li et al. [9] compare SVM with traditional neural networks for consumer credit data, with SVM being superior in terms of generalization performance. Huang et al. [6] also observe slight improvement of SVM over neural networks when used for credit rating analysis. Bellotti and Crook [3] report that SVM with linear or RBF kernel are successful when compared to more traditional methods. Using a very large data set of approximately 25000 credit card customers, however, they found no overwhelming superiority of SVM.

2 Extracting Information from Large Data Sets

In some applications like credit scoring, the number of cases or clients can be expected to further grow in the future, whereas the number of easily accessible features per client can be expected to remain approximately the same (barring the use of some new and controversial information on DNA sequences, on social networking, etc). Dealing with large enough N can pose problems to almost any classification procedure. In section 4 we describe credit client classification on a large empirical data set which already poses serious problems to SVM models which we successfully used in previous work for much smaller data sets (of type **A**, say).

Apart from seeking to alleviate numerical problems with large data sets, one may wonder whether the information from such large credit client data sets can be "compressed", possibly leading to new models on such compressed

data, which are comparable to the performance of models on the much smaller empirical data sets just mentioned.

A possible choice for the compression step is clustering and using cluster prototypes as the new training examples for SVM models. Clustering itself comes in many variants (see e.g. [7]) employing a multitude of procedures for selecting distances, similarities and other relations between pairs and other subsets of points. By selecting adequately, one can exploit knowledge about geometries and other characteristics of data sets and improve clustering results substantially (e.g. [2]).

Without having reliable sources of such knowledge for credit scoring data, one would be inclined to use in a first approximation a simple and fast clustering procedure. Hence, in the sequel, the method of choice will be k-means clustering, an ubiquitous method in many statistical computing resources.

As a general strategy one can attempt to increase or to stabilize the performance of classifiers on large data sets by reducing (or compressing) the number of training examples and, possibly, by adding more derived input features to the m primary inputs, and / or by modifying the representation of inputs (the first and the last will be discussed in the sequel), the basic idea of which is summarized in figure 1. As a first series of exploratory experiments we compute a large number of

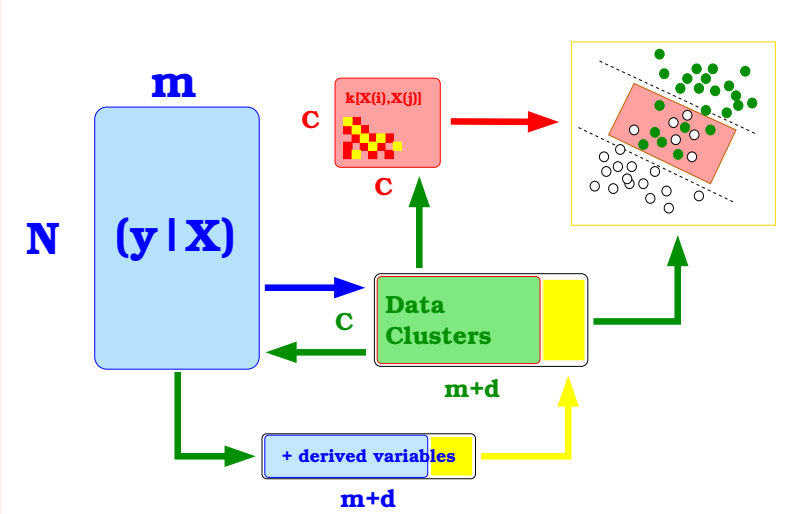

Fig. 1. Reduce and augment your data set. Reducing from N cases (clients) to $m < C < N$ prototypes of client clusters. Possibly enhance input feature information by adding some dimensions d (such that $m + d < N$), containing some aggregate or selective information from the respective cluster. A smaller $C \times C$ kernel matrix results for the SVM classifier.

3 Cluster-based SVM Trained on a Small Data Set

As a first series of exploratory experiments we compute a large number of cluster based SVM models. In this set of experiments, we choose a data set of type **A** from the same loan company, having a much smaller number of clients but also a larger number of client features ($m = 40$). For this data set we know from our previous work [10][11], which validated SVM models (using which kernels) are best when training is done in the usual case-by-case way.

Now we can describe a regime of clusterings for which one may obtain classification performance similar to that of models trained case-by-case. Keeping in mind that our credit scoring data uses binary class labels (i.e. "defaulting" and "non-defaulting" credit clients) we can use two variants:

- Cluster the two classes separately. Train the SVM on the labeled cluster prototypes. Perform cross-validation.
- Cluster the data without disclosing class memberships. Label the resulting cluster prototypes by the majority of the labels of the respective cluster members. If the majority is not surpassing a threshold (of 65% for one class, say) then unset the label. A semi-supervised classification problem results [8]. Cross-validation is not possible in this variant.

In addition, both variants can use a holdout sample for validation, which, however, may be a much better (less biased) procedure for a very big data set. In the first set of experiments we will use the second variant.

If clustering without using labeled subsets leads to good models, then using exactly two clusters would "solve" the entire classification problem. This is however not the case with our data set. At the other extreme, a number of clusters c approaching N is virtually the same as training case-by-case. Without using cross-validation, with most nonlinear kernels, overtraining inevitably results. For this actually to occur for large c we will use the RBF-kernel for all SVM models build on the $2 < c < N$ cluster prototypes.

Most importantly, in the present context of $N = 658$ credit clients, we have approximately equal distribution of positively (defaulting) and negatively (non-defaulting) labeled clients, a situation in which comparing misclassification rates over the different models makes perfect sense (this will change, however, for later experiments on other data sets). Figure 3 depicts the sorted misclassification errors of batches of 200 SVM-models trained on 50,60,80,110, and, respectively, 120 cluster prototypes. The 200 SVM models within a batch (fixed number of cluster prototypes) result from clusters of prescribed cluster number size produced by k-means and restarted randomly 200 times. This leads in general to different local minima of the clustering objective function and to a variation of the cluster structure (assignments of single cases to a cluster). As the number of cluster prototypes compared to the total number of cases is relatively large, this variation within batches is high (view the respective models within batches along each curve sorted by classification error). Hence such a cluster based approach is not to be recommended for

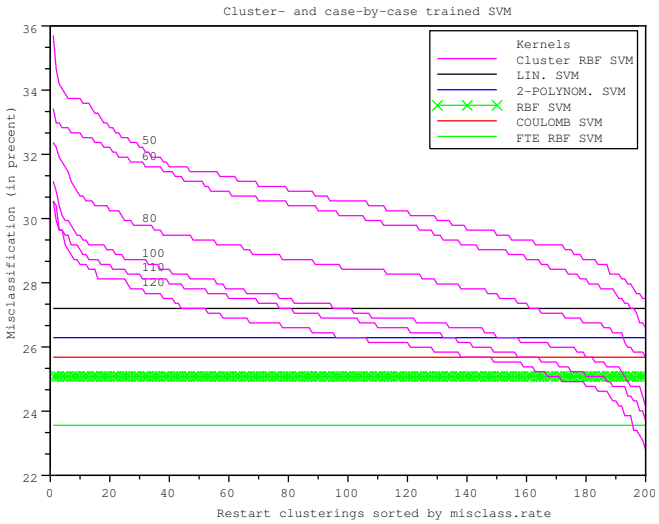

Fig. 2. Misclassification error over several batches of experiments with different numbers $C = 50, 60, ...$ of prototypes (the number beneath each curve). Every batch contains 200 runs (i.e.trained SVM with RBF kernel). The runs within a batch differ by the clusterings (assignment of cases to clusters) produced by randomly restarting k-means. The curves depict the sorted misclassification error over the 200 runs, respectively.

such type **A** data sets. But the important part to observe here is that models on compressed information (cluster based SVM) can surpass misclassification errors of the best models trained with all information available.

We note here that validation of the cluster based SVM models is not completely fair but may still be quite reasonable, at least for models with fewer cluster prototypes. Note, k-means is applied here to unlabeled data and that cluster prototypes resulting from the k-means algorithm are labeled ex-post by a majority vote (counting how many cluster members are of the same class) and by unlabeling "weak majorities" or "ties", which leads to a semi-supervised variant of SVM. Such a variant tries to label the unlabeled cases during the training phase. Validating semi-supervised models in a manner completely analogous to that of supervised models is computationally more involved and is part on ongoing research.

Figures 3, 4, and 5 illustrate the potential of using cluster based SVM for scoring credit clients but also remind us of the very strong interference between the restart effect of k-means on the resulting cluster structure which here is rather strong, even compared to the impact of rather large variations of the number of prescribed cluster prototypes on cluster structure. We now

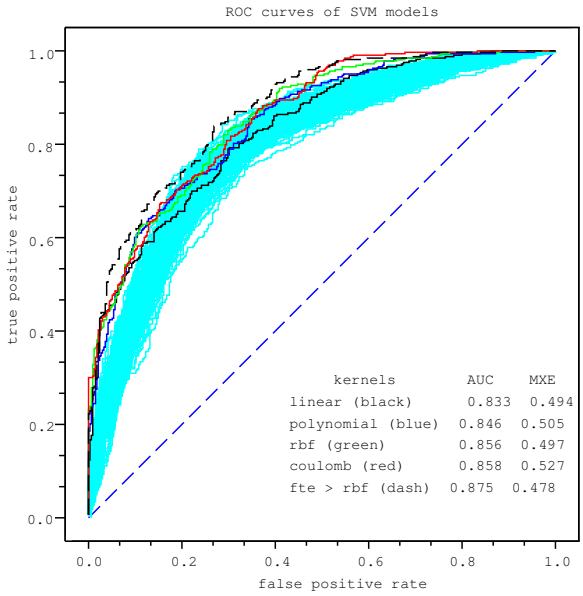

Fig. 3. ROC curves of base classifiers (trained in the ordinary case-by-case fashion) and ROC curves of a batch of classifiers trained on the cluster prototypes. The base classifiers are identified by their SVM kernels (see legend). One-leave-out validation error forecasts against true labelings are used for the computation of ROC curves / AUC (area under ROC) and MXE (mean cross entropy) of the base models.

proceed to examine how this translates to a much larger data set which also differs with regard to input features.

4 Related Experiments on a Large Data Set

We are now extending our experiments from the small data set to a much bigger data set to be used in the remaining sections. This credit scoring data is extremely unbalanced with respect to the number of defaulting and non-defaulting credit clients (a ratio of about 1:38), deeming standard measures as accuracy (or misclassification rate) impractical. A more complex measure is the ROC (Receiver operating characteristics) curve which offsets true and false positive rates of a classification model using model predictions against reference output (i.e. the true labels of the data set). The ROC area is defined as follows: Let $S(x)$ be a decision function (e.g. SVM) with continuous output. Then a cut-off value C can be used to determine the separation of credit clients into good and bad risk classes, with:

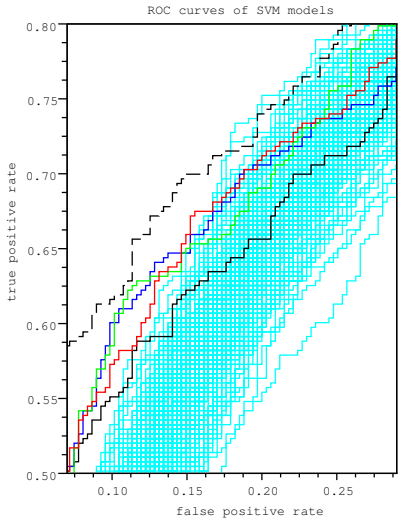

Fig. 4. Zooming a rectangle from the left hand side of figure 3 shows the ROC curves of the 200 models of a batch and the ROC curves of the reference models.

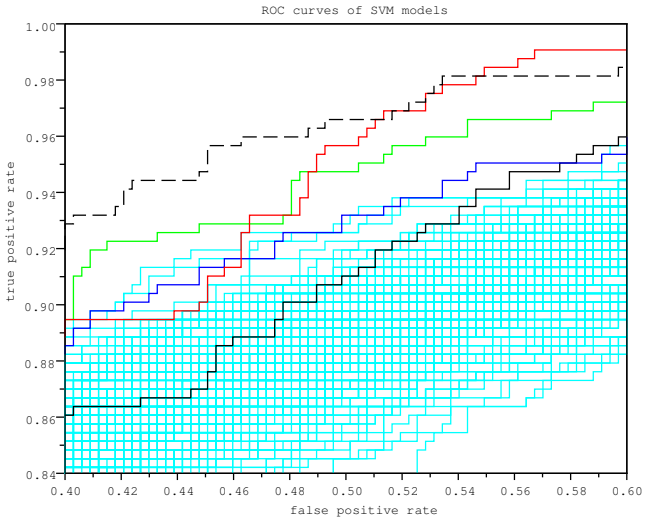

Fig. 5. Zooming a rectangle from the right hand side of figure 3 shows ROC curves surpassing those of the reference models.

$$S(x) < C \quad \longrightarrow \quad \text{classify as "good" client}$$
$$S(x) > C \quad \longrightarrow \quad \text{classify as "bad" client}$$

Subsequently, the **hit rate** (HR) and the **false alarm rate** (FAR), with given C, can be calculated as

$$HR(C) = \frac{H(C)}{H(C) + M(C)} \quad \text{and} \quad FAR(C) = \frac{F(C)}{F(C) + R(C)}, \quad \text{with}$$

$H(C)$ = Bad rejected $\qquad$ $F(C)$ = Good rejected
$M(C)$ = Bad accepted $\qquad$ $R(C)$ = Good accepted.

The graph using HR and FAR as coordinates for every possible cut-off C is known as Receiver Operating Characteristics (ROC) curve. The area between the "pure chance" diagonal (where the hit rate is equal to the false alarm rate for any given C and the classification function provides no useful information) and the ROC curve is a measure for the cut-off independent classification power of the decision function $S(x)$.

Here we perform a set of five experiments, which build SVM models on an increasing set of cluster prototypes computed by the same k-means clustering algorithm used in the previous section. Apart from using a much bigger data set, the present experiments differ in another important aspect from the previous ones of section **??**. The clustering is done separately on the sets of negatively and positively labeled examples, hence endowing the clusters with unambiguous labels which amounts to an extreme case of constrained clustering (a weaker version would be to enforce or to prevent same class membership of a small number of clients, etc.). These experiments are compared to another experiment, which simply samples 1000 clients randomly, but with the provision of being subdivided into approximately equally sized "classes" with opposite labels. These SVM models are also retrained using ten different such samples.

The outcome of these experiments is depicted in figure 6. In terms of numbers of cluster prototypes for SVM model, in our experiments 50,100, 200, 400, and respectively, 800, there is a steady increase in AUC (Area under the ROC curve) with moderate variation over the k-means algorithm restarts. Note, that less cluster prototypes then random data samples are needed in order to achieve a given AUC level, which is an argument in favor of the cluster based approach using ROC-based criteria for very large (and strongly asymmetric) credit client data sets.

Figure 7 depicts the ROC curves of the experiments from figure 6 of the cluster based SVM. The five rows of plots from figure 7 refer to 50, 100,200,400, and, respectively, 800 cluster prototypes, which are fed as training examples to the respective SVM models (using the RBF kernel only, with the same hyperparameters as in the previous experiments). The variability within the models is caused by restarting the k-means clustering algorithm, which leads

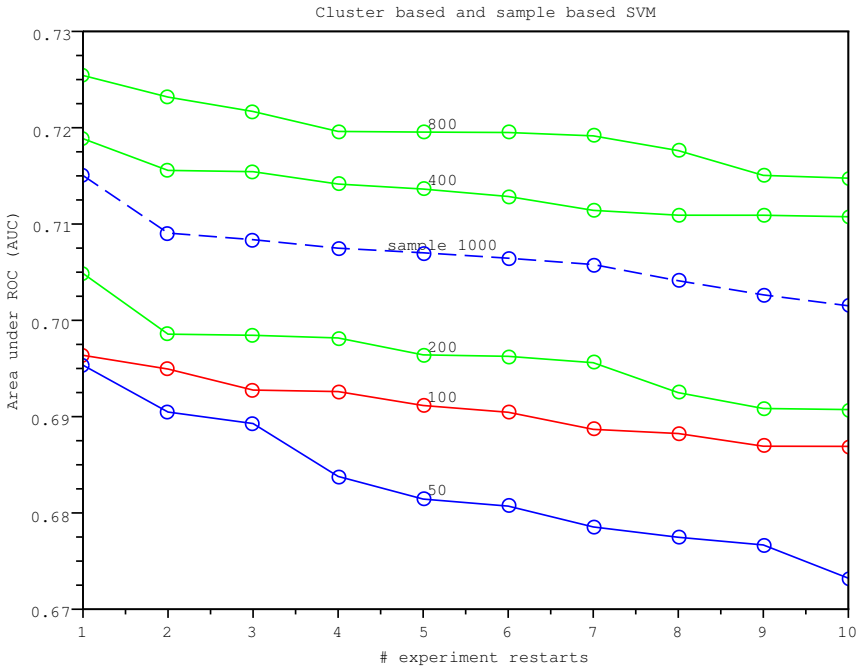

Fig. 6. Comparative Area under the ROC curve for five experiments with increasing number of clusters for SVM models and an experiment for randomly sampled cases for SVM models.

to slightly different local minima for each restart. Note that the variability is less then the one observed with the smaller data set from the previous section. This is attributable to the fact that the number of cluster-prototypes is now much less than the total number of training points. While the Lhs plots of figure 7 show the ROC of the two model from the respective experiment set which have maximum and minimum AUC respectively, the Rhs plots are depicting all ROC curves of the experiment but constrained to a small window of the Lhs ROC plot. We observe a weak tendency of decreasing variability of the ROC curves for increasing number of clusters. Interference of the restart effect from k-means with the

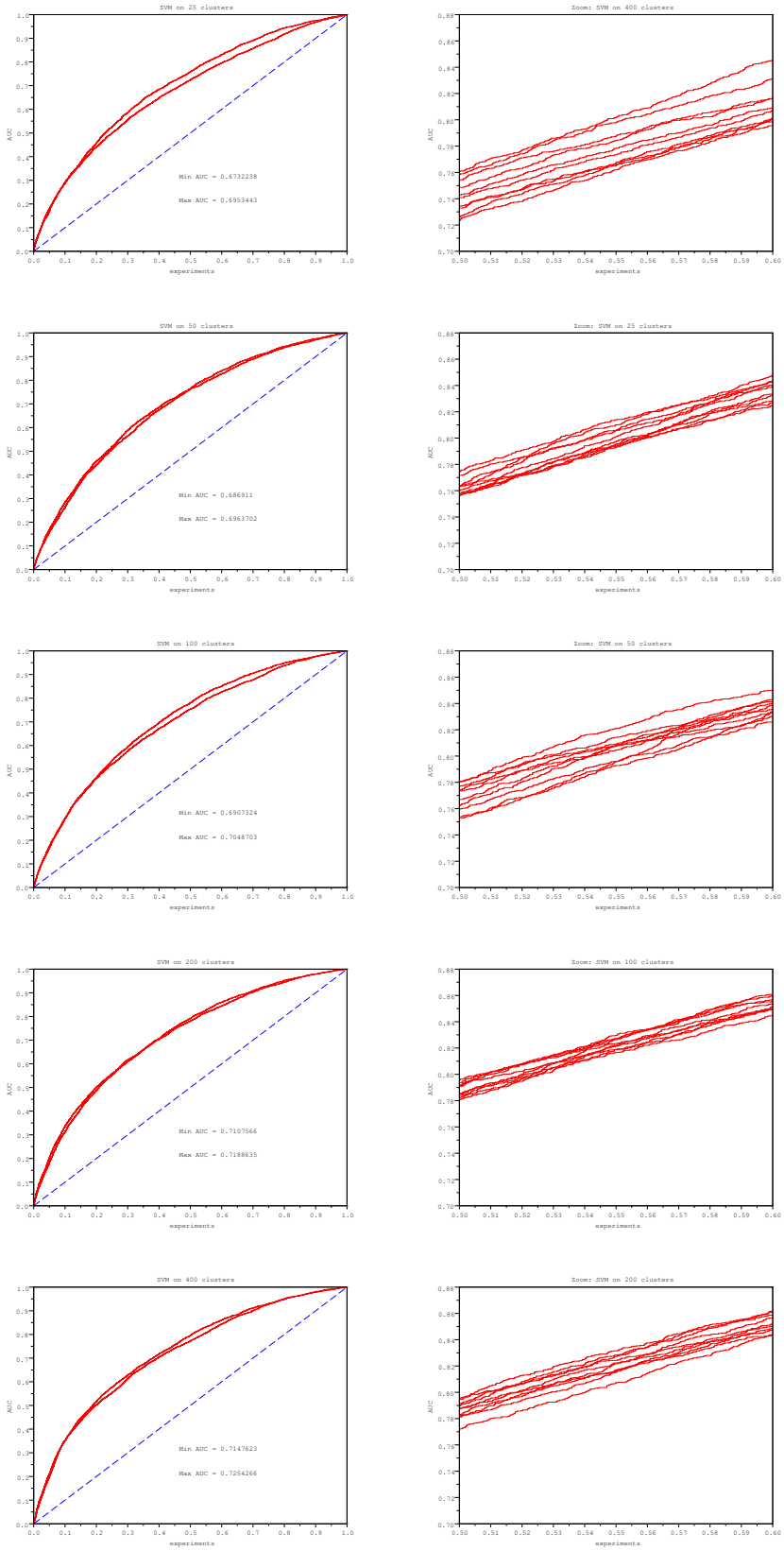

Fig. 7. Comparative Area under the ROC curve for five experiments with increasing number of clusters for SVM models and an experiment for randomly sampled cases for SVM models. Lhs: ROC curve with maximum and minimum AUC from experiment. Rhs: All the ten ROC curves in a small zoomed window.

5 Cluster Representation of Large Data Sets

The data set used consists of 139951 clients for a building and loan credit. There are twelve variables per client of which eight are categorical (with two up to five categories) and four are quantitative. The default rate is 2.6% (3692 out of 139951). The data set is large and in general not (or hardly) feasible for standard SVM. Furthermore, extremely unequal class sizes will make classification difficult. Former experiments suggest using a small (with equal sized classes) sample for SVM model building. This leads to mediocre out–of–sample classification results for the large data set [11]. In this work, an alternative way of dealing with such large data sets will be shown: First, a cluster analysis is performed and then, based on cluster representations, the SVM is estimated. This leads to several open questions: (*i*) which (supervised or unsupervised) cluster procedure has to be used, (*ii*) what number of clusters to choose, (*iii*) how to represent the clusters, and (*iv*) how to label the clusters. Advantages from a preprocessing cluster analysis are a massive downsizing of the large data set of N cases to a much smaller set of $n \ll N$ clusters. These cluster solutions may include a weighting scheme, e.g. using a balanced number of "good" and "bad" credit client clusters. If there is noise in the data it will be averaged out through clustering. Clusters can be represented as *symbolic objects*. It will be shown, that symbolic object coding is a comprehensive way to preprocess credit client data that easily can incorporate mixed (i.e. categorical *and* quantitative) variable types.

6 Symbolic Objects

In contrast to "classical" data analysis where elements are individuals with single variable outcomes (*"first degree objects"*), in symbolic data analysis elements, respectively *objects* in general, are *classes of individuals* or *categories of variables* that usually will be described by more than one outcome per variable (*"second degree objects"*) [5]. Clusters of credit clients can be seen as such symbolic objects with e.g. an interval representation of the amount of credit that was given to the cluster members. However, a special data description is needed to represent the variable outcomes of a symbolic object. A complete overview of symbolic variable types can be found in [4]. In the present work the clusters (the symbolic objects) are described by *modal variables* where categories or intervals (of categorical or quantitative) variables appear with a given probability. Each variable is represented by a vector of probabilities, e.g. $X_u = \{\eta_{u1}, p_{u1}; \ldots; \eta_{us}, p_{us}\}$ where outcome η_{uk} (of object u relating to variable X with $k = 1, \ldots, s$ outcomes) is given with a probability p_{uk} and $\sum_{k=1}^{s} p_{uk} = 1$. Outcomes η_{uk} may either be categories or (non–overlapping) intervals. Therefore, categorical as well as quantitative variables are represented by a vector of *standardized* values that are strictly between 0 and 1. In the framework of symbolic data analysis much work has been done regarding

the algebra of symbolic variables, including descriptive statistics, dependency measures and distance functions as well as diverse multivariate data analysis approaches [4].

7 Symbolic SVM

In a first step, the initial data set of 139951 cases is divided into "good" and "bad" credit clients. Subsequently, unsupervised k–means clustering is performed, extracting 50 clusters from the "good" and "bad" classes respectively. For each cluster the relative frequency of all categories per variable is recorded. Quantitative variables are divided into four equally sized intervals (with quartiles of the full variable range as interval borders). Relative frequencies for these intervals are also recorded. Finally, the *symbolic data set* consists of 100 objects (50 "good" and 50 "bad" clusters) with 43 inputs between zero and one. This symbolic data set is given as input to a linear SVM. As a first result, symbolic SVM divides the data set into "good" and "bad" risk classes without any training error. The leave–one–out error is 16% (see table 1). Moreover, SVM divides the data set into *typical* and *critical* regions [10]. Typical patterns do not affect the separating function and are easy to predict. Critical patterns (the "support vectors") are close to the border between separated classes and can hardly be assigned correctly to the respective true classes. The critical region of the linear SVM consists of 25 cluster representatives, whereas the typical region includes the remaining 75 cluster representatives. The relation between the number of typical and critical credit client clusters is a measure for the *predictability* of the whole data set.

8 Classification

The symbolic SVM can be used to describe the large data set by means of cluster regions. But can it also be used to predict the behaviour of new credit clients? To this purpose, we introduce *relative frequency coding* for the large data set. Each credit client is assigned to a category (or an interval) with a probability of either zero or one. Given this structure, the whole data set may be classified using the symbolic SVM that was estimated before. In order to get a true out–of–sample performance, the large data set was divided into a training set (consisting of two thirds of the credit clients, i.e. $N_T = 93391$) and a validation set (consisting of one third of the credit clients, i.e. $N_V = 46560$). The training set was clustered and used for SVM building. The resulting separating function is then used to predict the client behaviour on the *relative frequency coded* validation set.

SVM models with two different kernels are estimated by: (1) linear SVM and (2) SVM with RBF kernel. One resulting ROC curve for the validation set can be seen in figure 8. The ROC areas and their 95% confidence intervals (CI)

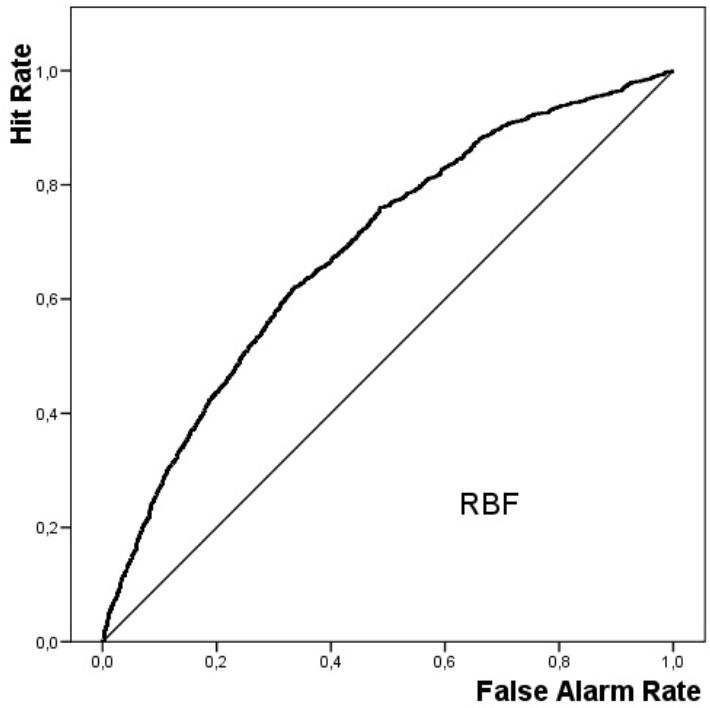

Fig. 8. ROC curve for SVM with RBF kernel. SVM is trained with the small cluster based data set ($n = 100$) and evaluated on the large credit client validation set ($N = 46560$). Area under ROC curve is 0.686 (see table 1).

are reported in table 1. Both models show competitive classification results with the RBF kernel outperforming the linear one, but not in significant way. Former experiments with the large data set include estimating classification functions in a more traditional way, by using a smaller ($n = 658$) hold–out–sample of credit clients with equally sized "good" and "bad" risk classes. Logistic regression and linear discriminant analysis were used as benchmark models for several SVM with different kernels (for detailed results see [11]). However, as outlined in table 1, it is evident that these classification results are worse in terms of ROC areas than the present approach of symbolic SVM with cluster based input.

9 Conclusions

Our exploratory experiments with a small and a very large credit client data set indicates that cluster based SVM models are a feasible approach for successfully describing and predicting credit clients. While the small data set

	Cluster based SVM		Sample based SVM		Traditional Methods	
	Linear	RBF	Linear	RBF	LDA	LogReg
Training						
No. of training cases	100	100	658	658	658	658
No. of SVs	25	51	357	431	–	–
Training error	0%	0%	22.64%	10.94%	23.86%	22.49%
L–1–o error	16%	12%	27.20%	25.08%	27.66%	27.51%
Forecasting						
ROC (Full Set)						
($N = 139951$)	0.656	0.675	0.583	0.576	0.582	0.584
[95% *CI*]	±0.09	±0.09	±0.11	±0.11	±0.10	±0.10
ROC (Validation)						
($N = 46560$)	0.660	0.686	–	–	–	–
[95% *CI*]	±0.15	±0.15	–	–	–	–

Table 1. Classification performance and ROC areas of SVM evaluated on sample and cluster based training subsets. Results are compared to traditional methods.

holds a very high variability of cluster structures within a batch of experiments with a fixed number of clusters this effect reduces substantially when using cluster based SVM on our very large data set. Furthermore, symbolic coding of clusters as symbolic objects is introduced, which enables more complex representation of cluster information. This leads to competitive classification results which are superior to former experiments with more traditional sample based SVM. We also note that model building process using cluster based SVM is a suitable way of treating such large and asymmetric credit client data. Further work will concentrate on more appropriate validation procedures and on exploring more complex symbolic cluster encodings.

References

1. BAESENS, B., VAN GESTEL, T., VIAENE, S., STEPANOVA, M., SUYKENS, J. and VANTHIENEN, J. (2003): Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54, 627–635
2. BASU, S. DAVIDSON, I. WAGSTAFF, K.L.: (2009): Constrained Clustering. Advances in Algorithms, Theory, and Applications. Data Mining and Knowledge Discovery Series, CRC Press, Chapman Hall 2009
3. BELLOTTI, T. and CROOK, J. (2008): Support vector machines for credit scoring and discovery of significant features. *Expert Systems with Applications, to appear.*

4. BILLARD, L. and DIDAY, E. (2006): *Symbolic Data Analysis.* Wiley, New York.
5. BOCK, H.-H. and DIDAY, E. (2000): *Analysis of Symbolic Data: Exploratory Methods for Extracting Statistical Information from Complex Data.* Springer, Berlin.
6. HUANG, Z., CHEN, H., HSU, C.-J., CHEN, W.-H. and WU, S. (2004): Credit Rating Analysis with Support Vector Machines and Neural Networks: A Market Comparative Study, in: *Decision Support Systems (DSS)*, 37(4), 543-558
7. JAIN, A.K., MURTY, M.N. and FLYNN, P.J. (1999): *Data Clustering: A Review*, in: ACM Computing Surveys, Vol. 31, No. 3, S. 264-323
8. JOACHIMS, T. (2002): *Learning to classify text using support vector machines: methods, theory, and algorithms.* Kluwer, Boston.
9. LI, S., SHIUE, W., HUANG, M.H. (2006): The evaluation of consumer loans using support vector machines. *Expert Systems with Applications, 30, 4, 772–782.*
10. SCHEBESCH, K.B. and STECKING, R. (2005): Support vector machines for credit applicants: detecting typical and critical regions. *Journal of the Operational Research Society, 56(9), 1082-1088.*
11. STECKING, R. and SCHEBESCH, K.B. (2006): Variable Subset Selection for Credit Scoring with Support Vector Machines. In: Haasis, H.-D., Kopfer, H. and Schönberger, J. (Eds.): *Operations Research Proceedings 2005.* Springer, Berlin 251–256.