A Case Study of Core Vector Machines in Corporate Data Mining

Stefan Lessmann, Ning Li, Stefan Voß
Institute of Information Systems, University of Hamburg

Abstract

The core vector machine (CVM) has been introduced as an extremely fast classifier which is demonstrably superior to standard support vector machines (SVMs) on very large datasets. However, only limited information regarding the suitability of CVM for supporting corporate planning is available so far. In this paper, we strive to overcome this deficit. In particular, we consider customer-centric data mining which commonly involves classification in medium-sized settings. CVMs are compared to SVMs within the scope of an empirical benchmarking study to clarify whether previous findings regarding the competitiveness of CVMs generalize to business applications. To that end, representative real-world datasets are employed. In addition, the study aims at scrutinizing the behavior of CVM during model selection. Following a standard grid-search based approach we find some evidence for CVM being more sensitive towards parameter settings than SVMs.

1. Introduction

Data mining has become an important tool to support customer-centric planning tasks in, e.g., response modeling [5, 10], customer attrition analysis [13, 18], credit scoring [23, 28] or fraud detection [16, 30]. Such applications are commonly approached by means of supervised classification and SVMs have proven their suitability for respective decision problems [4, 8, 19, 22].

Recently, Tsang et al. [24, 25] have introduced the CVM as a novel classifier. CVMs possess substantial similarities with traditional SVMs but are more efficient for mining very large datasets. In particular, the quadratic program underlying SVMs is reformulated as a minimum-enclosing-ball problem which solution can be approximated by means of a fast, iterative algorithm. For example, CVMs have been shown to construct a classifier on datasets of up to five million examples and approximately 100 variables within seconds on contemporary hardware without sacrificing predictive accuracy [24]. This is an exceptional result and exceeds the computational capabilities of traditional SVMs by far. However, Tsang et al. [24, 25] demonstrate that the latter can be more efficient on small datasets. Consequently, the current body of knowledge regarding CVMs, e.g. [1, 2, 17, 24, 25, 26, 27], naturally raises question which method to apply in medium sized settings. Contributing towards answering this question from a perspective of corporate data mining is one of the aims of this paper.

Corporate data mining tasks commonly involve datasets of medium size. On the one hand, customer-centric data is collected in almost any business transaction due to extensive usage of information systems. On the other hand, the predominant approach to model customer behavior involves mapping one customer to one example, i.e. one record in the dataset to be mined. Therefore, the size of such datasets is naturally bounded by the number of a company's customers. In addition, the supervised learning paradigm imposes further constraints on the availability of useable training data by requiring detailed label information, i.e., a specific value for the dependant variable for each customer.

Therefore, the paper strives to appraise CVMs potential for customer-centric classification tasks. In particular, we conduct an empirical experiment to contrast CVMs and SVMs (as reference model) on representative datasets. Amending traditional measures of comparison like predictive accuracy and computational efficiency, a classifier's sensitivity towards parameter settings is considered as an additional quality indicator.

Both methods exhibit the same free parameters and thus require model selection techniques to determine suitable values. This task is predominantly approached by means of empirical procedures that repetitively evaluate different

1530-1605/08 $25.00 © 2008 IEEE

candidate values. Consequently, parameter sensitivity increases the number of evaluation and thereby the overall training time of the classifier. Furthermore, higher sensitivity elevates the risk of selecting a suboptimal setting which produces inferior out-of-sample accuracy. A standard argument within the corporate data mining community is that small deviations in predictive accuracy can have substantial financial consequences [4, 5, 8]. Therefore, a more robust method might be preferable despite computational inferiority.

To appraise the parameter sensitivity of CVM and SVM, we propose a worst-case analysis as well as an analysis based on the fourth statistical moment of the respective performance distributions. Following this approach, the model selection results presented here provide some evidence for CVMs being more sensitive towards parameter settings than SVMs. In other words, the latter may be considered appropriate even if constructing a single classifier on a given dataset turns out to be more time consuming than conducting the same task with the CVM.

The paper is organized as follows. We briefly review the basics of SVMs in Section 2 before discussing the reformulation considered in CVMs. Sections 3 and 4 present the empirical results of the benchmarking study, and conclusions are drawn in Section 5.

2. Classification algorithms

In the sequel, we review the theory of SVM- and CVM-based classification. Formally, the task of classification can be stated as follows: Let S be a dataset containing M examples, $S = \{(\boldsymbol{x}_i, y_i)\}_{i=1}^{M}$, where $\boldsymbol{x}_i \in R^N$ denotes an input vector and y_i its corresponding discrete class label. The goal of classification is to infer a predictive model, i.e. a classifier, $y(\boldsymbol{x})$ from S, which accurately predicts the class membership of novel examples. Here, we consider the case of binary classification where $y_i \in \{-1, +1\}$.

2.1. Support vector machines

SVMs can be characterized as linear classifiers. That is, predictions are based on a separation of the data by means of a linear hyperplane (1), with normal $\boldsymbol{w}$ and intercept b:

$$y(\boldsymbol{x}) = sign\left(\boldsymbol{w} \cdot \boldsymbol{x} + b\right). \tag{1}$$

For SVMs, the parameters, $\boldsymbol{w}$ and b, are determined by means of mathematical programming. Thus, the construction of the classifier, commonly referred to as classifier training, corresponds to solving the convex program (2) to optimality, whereby ξ_i represents a slack variable which is greater than zero only if a training example $\boldsymbol{x}_i \in S$ is misclassified.

$$\begin{aligned} &\min_{\boldsymbol{w},b,\xi} \frac{1}{2}\boldsymbol{w} \cdot \boldsymbol{w} + C\sum_{i=1}^{M}\xi_i \\ &s.t.:\ y_i((\boldsymbol{w} \cdot \boldsymbol{x}_i) + b) + \xi_i \geq 1,\ \ i = 1,...,M. \end{aligned} \tag{2}$$

Program (2) is inspired by statistical learning theory [29] and minimizes the sum of two terms which measure the distance between the examples of opposite classes which are closets to the hyperplane defined by $\boldsymbol{w}$ and b, referred to as the margin of separation, and the number of misclassifications, respectively.

The margin can be related to the model's capability of producing predictions that generalize to future data. Roughly speaking, SVMs strive to discriminate the training data accurately, i.e. without error, with a model as simple as possible, i.e. a model with large margin; see [9] for details. The parameter C allows controlling the trade-off between these two conflicting goals and has to be specified by the user prior to classifier training. Subsequently, we refer to C as the penalty parameter.

A mapping function is employed to produce more complex, nonlinear classifiers. That is, the classification model (3) is considered instead of (1), whereby φ is a nonlinear mapping that projects $\boldsymbol{x}$ into a higher dimensional feature space.

$$y(\boldsymbol{x}) = sign\left(\boldsymbol{w} \cdot \varphi(\boldsymbol{x}) + b\right). \tag{3}$$

By constructing the linear classifier in this nonlinearly transformed space, a nonlinear separation of the data in the input space R^N is obtained. Due to the structure of SVMs, it is not necessary to explicitly compute this transformation. Consider the dual of (2) and let α_i denote the Lagrangian multipliers:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} = \sum_{i=1}^{M}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{M}\alpha_i\alpha_j y_i y_j (\boldsymbol{x}_i \cdot \boldsymbol{x}_j) \\ &s.t. \quad \sum_{i=1}^{M}\alpha_i y_i = 0 \quad 0 \leq \alpha_i \leq C \quad \forall i. \end{aligned} \tag{4}$$

As the input data enters the dual (4) only in form of scalar products, a so called kernel func-

tion K (5) can be employed to compute the scalar products of the transformed vectors directly:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i) \cdot \varphi(\boldsymbol{x}_j). \quad (5)$$

Thus, the final SVM classifier is given as:

$$y(\boldsymbol{x}) = sign\left(\sum_{i \in SV} \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b \right), \quad (6)$$

where SV represents the set of support vectors, i.e examples with non-zero Lagrangian multipliers.

The kernel (5) defines a measure of proximity between examples in the transformed feature space. Integration of a kernel into (4) is straightforward and does not affect the overall algorithm. This may be considered a particular merit of the SVM classifier which leads to increased flexibility, e.g. by developing special purpose kernels for text or multi-media mining tasks or incorporating prior knowledge. However, the radial basis function (RBF) kernel (7) is most popular in practical applications of corporate data mining and has been shown to posses some desirable properties [15]. Therefore, this kernel is used later in the benchmarking experiment.

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma \left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|^2 \right). \quad (7)$$

2.2. Core vector machines

CVMs have initially been proposed in [24, 25]. Extensions to the task of support vector regression and classification with class-dependant penalties are proposed in [26, 27]. Furthermore, CVMs have been considered in conjunction with clustering algorithms [1] and multi-class classification [2]. A classifier closely related to the CVM has also been proposed in [17].

As solving (4) involves quadratic programming, SVM learning may become infeasible in large-scale settings when datasets comprise several hundred thousand examples. Observing that practical algorithms for SVM learning, e.g. [20], do not solve (4) to optimality but impose a tolerance parameter on the Karush-Kuhn-Tucker conditions, Tsang et al. [25] propose to reformulate (4) as an equivalent minimum-enclosing-ball (MEB) problem which solution can be approximated by means of a fast iterative algorithm using the concept of core sets [24].

Given a set of points, e.g. $\boldsymbol{x}_i \in S$, the MEB is defined as the smallest ball which contains all points. Let r denote the radius and c the center of a ball, the problem of finding an MEB in the feature space can be stated as follows (8):

$$\begin{aligned} &\min_{r,\boldsymbol{c}} \; r^2 \\ &s.t. \quad \left\| \varphi(\boldsymbol{x}_i) - \boldsymbol{c} \right\| \le r^2 \quad \forall i = 1,...,M. \end{aligned} \quad (8)$$

The corresponding dual is given as:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \; \sum_{i,j=1}^{M} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &\quad - \sum_{i=1}^{M} \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_i) \\ &s.t. \quad \sum_{i=1}^{M} \alpha_i = 1 \quad \alpha_i \ge 0 \quad \forall i = 1,...,M, \end{aligned} \quad (9)$$

If:

$$K(\boldsymbol{x}_i, \boldsymbol{x}_i) = \kappa, \text{ a constant}, \quad (10)$$

one may discard the second term in the objective to obtain the final MEB problem (11).

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} \; \sum_{i,j=1}^{M} \alpha_i \alpha_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) \\ &s.t. \quad \sum_{i=1}^{M} \alpha_i = 1 \quad \alpha_i \ge 0 \quad \forall i = 1,...,M, \end{aligned} \quad (11)$$

Note that (10) holds for many practical kernel functions, including the RBF kernel. However, a generalization of the CVM [27] does not require this constraint anymore and enables arbitrary kernels.

As is shown in [2, 24, 25], a slight modification of the original SVM program (2) yields a dual similar to (11). In particular, considering the L2-norm of the slack variable, in other words using a squared-error loss function, produces the dual (12), with δ_{ij} being the Kronecker delta:

$$\begin{aligned} &\max_{\boldsymbol{\alpha}} = \sum_{i,j=1}^{M} \alpha_i \alpha_j \left(y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + y_i y_j + \frac{\delta_{ij}}{C} \right) \\ &s.t. \quad \sum_{i=1}^{M} \alpha_i = 1 \qquad \alpha_i \ge 0 \qquad \forall i. \end{aligned} \quad (12)$$

Now, to obtain (11), set:

$$\tilde{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + y_i y_j + \frac{\delta_{ij}}{C} \quad (13)$$

Hence, redefining the kernel by (13) allows formulating the SVM with L2-loss as a MEB problem.

The computational advantage of solving the MEB problem with an approximation algorithm

stems from the concept of core sets. Given a set of points $\boldsymbol{x}_i \in S$, a subset $Q \subseteq S$ is a core set of S if an expansion by a factor $(1+\varepsilon)$ of its MEB contains S [24], where ε is a small positive number. Tsang et al. [25] employ the algorithm of Bădoiu and Clarkson [3] to obtain such an ε-approximation of (11): Let $B_t(\boldsymbol{c}_t, r_t)$ denote the MEB of the core set Q at iteration t. Then, the algorithm adds to Q the furthest point outside the ball $B(\boldsymbol{c}_t, (1+\varepsilon)r_t)$. This step is repeated until all points in S are covered by $B(\boldsymbol{c}_t, (1+\varepsilon)r_t))$; see [24] for details.

CVMs efficiency on large datasets can be attributed to the fact that the size of the final core set depends only on ε but not on M or N [25].

The calculation of class predictions using the CVM differs from (6) only in the sense that the modified kernel $\tilde{K}$ is considered instead of K which also encodes label information y_i; namely:

$$y(\boldsymbol{x}) = sign\left(\sum_{i \in Q} \alpha_i \tilde{K}(\boldsymbol{x}_i, \boldsymbol{x}) + b\right). \qquad (14)$$

Note that we can always remove the sign-function in (1), (3), (6) and (14) to obtain a continuous prediction which represents the confidence of the classifier [29].

3. Comparing the computational efficiency of CVM versus SVM

One motivation for evaluating the CVM as a candidate technique for business classification is its remarkable computational efficiency. Therefore, we begin the empirical evaluation with a small runtime comparison of CVMs versus SVMs to replicate the results of [24, 25] in a setting of corporate data mining. In particular, we consider a case of direct marketing in the publishing industry. The respective dataset represents a marketing campaign aiming at cross-selling an additional magazine subscription to customers of the publisher. Each customer is characterized by 95 numerical as well as categorical attributes and 300,000 examples are given. The binary target variable indicates if a contacted customer has responded to the campaign, i.e. subscribed to one or more periodicals; see [10] for details.

Increasing numbers of examples are randomly sampled to scrutinize the evolution of training times. The LibSVM library [7] is employed as SVM implementation and both classifiers are configured with their respective default parameters. Table 1 depicts the results of this comparison in terms of training time and number of identified core vectors and support vectors, respectively.

Table 1: Efficiency comparison of CVM versus SVM

	Training set size in 1000 examples				
	60	120	180	240	300
	Runtime in sec.				
CVM	52	60	78	85	96
SVM	292	1,736	4,910	7,858	14,101
	Number of core/support vectors				
CVM	1,150	1,368	1,612	1,735	1,793
SVM	2,263	4,179	6,319	8,226	10,139

The results confirm previous findings [24, 25] and further emphasize CVMs efficiency on large datasets. Besides significantly lower training times, the number of core vectors is considerably smaller than the respective figure for SVM. Therefore, CVMs are significantly faster at classifying novel examples than SVMs for this task; see also (14) and (6), respectively.

4. Comparing predictive accuracy and parameter sensitivity of CVM versus SVM

4.1. Experimental setup

The previous results demonstrate that 60,000 training examples may suffice to give CVMs a computational advantage over SVMs. Therefore, subsequent experiments consider smaller datasets to enhance our understanding when to use which classifier. To that end, we employ four datasets from the Data Mining Cup, which is an annual competition organized by prudsys AG [21]. The considered data stems from the years 2000 to 2002 as well as 2005 (DMC00, DMC01, DMC02, DMC05) and represent classification tasks in direct marketing, churn prediction and fraud detection; see [19] for details. We deem these datasets representative for the domain considered here and summarize their characteristics in Table 2.

Table 2. Dataset characteristics

	DMC00	DMC01	DMC02	DMC05
#Train	10,000	10,000	10,000	30,000
#Test	28,890	18,128	10,000	20,000
%Pos	5.7%	50%	10%	5.9%
%MV	5.6%	22.6%	24%	84%
#CA	24	9	13	84
#MA	19	24	19	8

#Train/#Test: the number of records used during building/evaluating the classification model.
%Pos/%MV: the percentage of class 1 records and records that contain at least one missing value, respectively.
#CA/#MA: the given number of categorical and numerical attributes within the datasets.

The partitioning of examples into training/test records is taken from the particular challenge. With respect to the study's focus on predictive performance, standard pre-processing techniques are utilized; e.g., mean replacement of missing values, normalization to zero mean and standard deviation of numerical variables as well as dummy-variable-based encoding of categories; see, e.g., [10].

A model's predictive performance is measured by means of the area under a receiver-operating-characteristics-curve (AUC) [6]. The AUC is a general indicator of predictiveness and is selected because of its robustness towards imbalanced class distributions. Class imbalance is present in DMC01, DMC03 as well as DMC05, and may be considered characteristic for most customer-centric decision problems. In particular, AUC appraises the ranking capabilities of a model, i.e. the probability that a classifier ranks a randomly selected positive example higher than a randomly selected negative one and is thus equivalent to the Wilcoxon test of ranks [11].

The tolerance parameter ε is not considered in this study but left on its default setting for the CVM and the SVM. This leaves two free parameters that have to be specified prior to applying a CVM and a SVM classifier, respectively. These are the penalty parameter, C, as well as the width of the RBF kernel function, γ. We organize parameter determination as a grid-search over candidate values of $\log_2(C)$ = [-5,-3,-1,…, 15] and $\log_2(\gamma)$ = [-15,-13,-11,…,1]; e.g. [12]. Each of the resulting 99 parameter combinations is evaluated by means of 10-fold cross-validation on the training set to estimate the predictive power of the resulting classifier. The best setting is retained and a respective classifier is constructed on the whole training set to predict the test set.

4.2. Results of the model selection stage

The primary objective of analyzing the detailed grid-search results is to appraise the risk of model misspecification when applying the novel CVM classifier. As the likelihood of selecting suboptimal parameter values increases with the classifier's parameter sensitivity, i.e. the variation in predictive accuracy induced by different parameter settings, we utilize the latter as a proxy of misspecification risk.

As a first step towards a deeper understanding of CVMs' behavior during model selection, we consider a worst-case perspective. In particular, we may ask how the worst possible CVM model, with respect to the abovementioned parameter grid, compares to the worst SVM model. This idea is implemented by drawing the sorted cross-validation based AUC estimates over all parameter combinations for CVM and SVM (Figure 1). Hence, the abscissa of Figure 1 gives the rank value of a particular parameter setting, whereby the best setting obtains the highest rank.

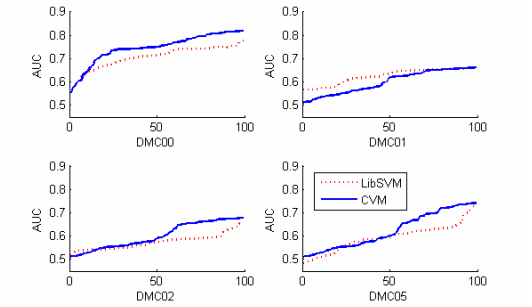

Figure 1. Ordered AUC for CVM and SVM per parameter setting

Figure 1 suggests that CVMs compare favorably to SVMs in the sense that an AUC estimate with given rank is commonly at the same level, or above, a respective SVM result. The situation is different on DMC01 where the ~70% least good parameter settings produce a lower AUC as in the SVM case. However, with respect to the ultimate goal of model selection, i.e. identifying promising parameter values, special consideration should be given to the top ranked parameters. Consequently, we may conclude that CVM is at least not inferior to SVM on the datasets employed here.

The best parameter values per dataset are reported in Table 3.

Table 3. Optimal parameter values per dataset and classifier by means of 10-fold cross-validation AUC

	DMC00	DMC01	DMC02	DMC05
	CVM classifier			
$\log_2(C)$	-1	5	1	-1
$\log_2(\gamma)$	-5	-11	-9	-7
AUC	0.82	0.66	0.68	0.74
	SVM classifier			
$\log_2(C)$	13	13	15	13
$\log_2(\gamma)$	-15	-15	-15	-15
AUC	0.78	0.66	0.66	0.73

Noteworthy, the parameters selected by SVM are more consistent and identical on three datasets. Considering CVM, higher variation of the penalty parameter C could be attributed to the fact that a L2-loss function is considered which might be more sensitive to outliers, see Section 2.2, whereas higher variation of γ is yet unexplained.

To gain further insight into CVMs' parameter sensitivity, Figure 2 depicts the distribution of AUC-values over the 99 parameter combinations per cross-validation fold and classifier across all datasets by means of box-plots. Datasets are ordered consecutively starting with DMC00 (first row).

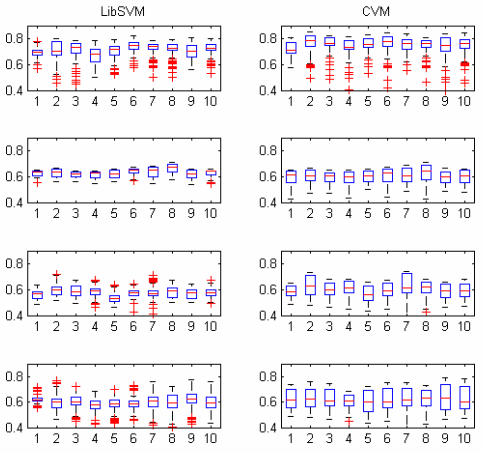

Figure 2. Distribution of AUC values per classifier and cross-validation partition

Clearly, both classifiers exhibit considerable variation which illustrates their parameter sensitivity and demonstrates the importance of model selection in general. For example, the median AUC value on DMC00 is 0.75 for CVM with a maximal (minimal) value of 0.81 (0.55) and respective figures of 0.71, 0.78 and 0.56 for SVM. Considering the results on the last three datasets, i.e. larger box height, one may speculate if CVMs' parameter sensitivity exceeds those of SVM. To further scrutinize this suspicion, we consider the fourth moment of the AUC distributions, the kurtosis, as depicted in Table 4.

Table 4. Kurtosis of AUC distributions per dataset and classifier

	DMC00	DMC01	DMC02	DMC05
CVM	1.155	-1.451	-1.525	-1.488
SVM	0.771	-0.956	-0.011	-0.108

The kurtosis is a measure of "peakedness" (or "tailedness") of a distribution compared to the normal distribution which has a kurtosis of zero [31]. The fact that CVMs' AUC distributions have smaller kurtosis on three out of four datasets indicates that the same parameter settings produce performance distributions with smaller peak and thus larger dispersion. This may be seen as evidence for the initial suspicion of CVM being more sensitive towards parameter settings.

A more elaborate approach, yet leading to the same conclusion, is as follows: Model selection strives to identify accurate models. Therefore, we may discard inappropriate parameter settings and focus on analyzing good results. This is done in Figure 3, which illustrates the development of kurtosis, when only the top N-percent of parameter combinations are considered. That is, we extract the 50%, 40%, etc. best parameter settings for both classifiers and compute the kurtosis for the resulting AUC distributions.

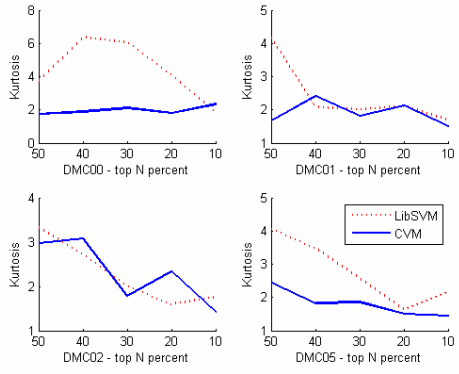

Figure 3. Kurtosis of upper percentiles of the AUC distribution across classifiers and datasets

The SVM shows demonstrably higher kurtosis on DMC00 and DMC05, whereas this pattern can be observed on DMC01 only up to the 40% best parameter settings. Mixed results are obtained on DMC02. Overall, the results provide some empirical evidence for CVMs' increased parameter sensitivity.

Note that the scope of the empirical evaluation requires distributing computations across multiple workstations with varying hardware configurations. Consequently, we refrain from presenting detailed runtimes for the model selection stage. However, we are able to report that CVM model selection consumes significantly more time than conducting the respective task for LibSVM for the considered datasets. For example, processing the 990 SVM models (99 parameter combinations * 10-fold cross validation) for DMC00 takes 93,778 sec., whereas CVM requires 496,098 sec. (The same hardware has been used for these two experiments, i.e. a Windows XP PC with 1.75Ghz CPU and 1GB RAM. As indicated by Tsang et al. [24, 25], SVM is more efficient for small sized problems because of sophisticated heuristics to speed-up classifier training. In addition, CVM tends to select a larger number of core vectors in such settings which, in turn, increases the time for classifier evaluation; e.g. the final SVM classifier for DMC00 includes 1,204 support vectors, whereas the CVM model comprises 6,880 core vectors. A similar yet less extreme pattern could be observed on DMC05. Combining this observation with the results of Table 1, it can be assumed that the CVM requires at least 40,000 to 50,000 examples to offer a computational advantage over the SVM.

4.3. Hold-out set results

Finally, Table 5 concludes the empirical comparison and depicts the predictive accuracy of the final CVM and SVM classification model on training and hold-out testing data. The results are roughly at the same level across both classifiers, with CVM giving slightly higher accuracy on DMC00. This confirms previous results of Tsang et al. [24, 25] regarding the competitive performance of CVM and demonstrates that they generalize to the datasets considered here. Furthermore, the overall experience with the CVM classifier in this study, as well as in previous experiments [1, 2, 24, 25, 27], further secures the initial conclusion of [24], "*that it* [CVM] *is as accurate as existing SVM implementations*" in terms of hold-out test set performance. However, due to amplified parameter sensitivity, model selection results might be less stable, leading to an increased risk of model misspecification. Although no case of predictive inferiority has been observed so far, this issue should be kept in mind before applying the CVM classifier.

Table 5. Training and test set results of the final classifiers by means of AUC

	DMC00	DMC01	DMC02	DMC05
	Final CVM classifier			
Train	0,82	0,66	0,68	0,74
Test	0,82	0,66	0,67	0,59
#CV	6,880	9,569	9,649	24,679
	Final SVM classifier			
Train	0,78	0,66	0,66	0,73
Test	0,79	0,66	0,66	0,59
#SV	1,204	8,264	2,037	3,591

5. Conclusions

Following an empirical research paradigm, we have evaluated a novel classification model, the CVM classifier, as a tool for corporate data mining. Our experiments replicate previous findings regarding the potential of CVMs and demonstrate that it is a promising approach for large-scale business classification tasks.

It is well known that the classification performance of a SVM model heavily depends upon a suitable selection of parameter values. Analyzing CVMs model selection behavior we have found some evidence for CVM being even more sensitive towards parameterization than SVM. In particular, parameter-induced performance variability of CVM exceeds that of a SVM classifier. Consequently, results of model selection might be less stable, leading to an increased risk of model-misspecification. However, no respective case has been observed empirically so far. On the contrary, we could replicate previous findings of CVM being at least competitive to SVM in terms of hold-out test set performance. Therefore, the question how severe practical applications are affected by slightly higher parameter variability requires further research. On the one hand, parameter sensitivity is not problematic as long as the employed model selection procedure, e.g. grid-search, selects "the right" configuration, i.e. parameter values that yield accurate hold-out predictions. On the other hand, training data used during model selection is always just a sample

and might give a biased picture of the stochastic process which has generated the data in the first place. In this sense, (higher) parameter dependency is undesirable. Furthermore, higher variability requires more extensive model selection, i.e. evaluating more parameter combinations, thereby decreasing CVMs computational advantage to some extend.

In this sense, we may conclude that CVM amend SVM and offer a capable alternative when the volume of the data to be processed prohibits application of the later. This is also evident from the fact that CVM can be considerably slower than SVM on smaller sized datasets [24, 25]. In medium-sized settings, users have to decide between both techniques. Our results suggest that the time for constructing a single classifier is a misleading indicator in such settings. Even if the size of the respective dataset suffices to give CVM a computational advantage over SVM, the former might still require a larger number of parameter evaluation to arrive at the same level of stability. Conversely, SVM facilitates using a coarser parameter grid and thereby regain some efficiency compared to CVM.

However, this does not depreciate the remarkable potential of CVMs. They enable classification in scenarios where the SVM can no longer be applied directly. One may object that it is not necessary to utilize all available data in large-scale settings but could employ SVMs in conjunction with sampling procedures. While true, we emphasize that each additional component, e.g. a sampling algorithm, adds to the overall complexity of the data mining process and thereby hinders a wider adoption in corporate practice.

As classification performance depends so heavily upon appropriate parameter values, the development of more sophisticated model selection procedures seems a promising field for future research. Substantial achievements have been made in the SVM community, e.g. by using gradient-based techniques [14]. On the other hand, this is the first study that considers CVM model selection in some detail. Gradient-based optimization of free parameters might be an option if they scale up to very large datasets where CVM unfold their full potential. Considering the approximate nature of the approach tuning heuristics like evolutionary algorithms appear to be another capable direction for future research.

Acknowledgments

The authors would like to express their gratitude to Ivor W. Tsang, James T. Kwok and Pak-Ming Cheung for making available the CVM executables. In particular we are grateful to James T. Kwok for continuous assistance and providing several valuable comments.

References

[1] S. Asharaf, M. N. Murty, and S. K. Shevade, "Cluster Based Core Vector Machine," *Proc. of the 6th IEEE Intern. Conf. on Data Mining*, Hong Kong, China, 2007, pp. 1038-1042.

[2] S. Asharaf, M. N. Murty, and S. K. Shevade, "Multiclass Core Vector Machine," *Proc. of the 24th Intern. Conf. on Machine Learning*, Corvallis, OR, USA, 2007 (to appear).

[3] M. Bădoiu and K. L. Clarkson, "Optimal Core Sets for Balls," in *Proc. of the DIMACS Workshop on Computational Geometry*. Piscataway, NJ, USA, 2002 (http://cm.bell-labs.com/who/clarkson/coresets1.pdf).

[4] B. Baesens, T. Van Gestel, S. Viaene, M. Stepanova, J. Suykens, and J. Vanthienen, "Benchmarking state-of-the-art classification algorithms for credit scoring," *Journal of the Operational Research Society*, 54(6), pp. 627-635, 2003.

[5] B. Baesens, S. Viaene, D. Van den Poel, J. Vanthienen, and G. Dedene, "Bayesian neural network learning for repeat purchase modelling in direct marketing," *European Journal of Operational Research*, 138(1), pp. 191-211, 2002.

[6] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognition*, 30(7), pp. 1145-1159, 1997.

[7] C.-C. Chang and C.-J. Lin, "LIBSVM - A Library for Support Vector Machines," 2001. (www.csie.ntu.edu.tw/~cjlin/libsvm)

[8] K. Coussement and D. Van den Poel, "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques," *Expert Systems with Applications*, 34(1), pp. 313-327, 2008.

[9] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other Kernel-based Learning Methods*. Cambridge: Cambridge University Press, 2000.

[10] S. F. Crone, S. Lessmann, and R. Stahlbock, "The impact of preprocessing on data mining: An evaluation of classifier sensitivity in direct marketing," *European Journal of Operational Research*, 173(3), pp. 781-800, 2006.

[11] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, 27(8), pp. 861-874, 2006.

[12] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," Working paper, *Department of Computer Science and Information Engineering, National Taiwan University*, 2003.

[13] Y. Hur and S. Lim, "Customer Churning Prediction Using Support Vector Machines in Online Auto Insurance Service," *Proc. of the 2nd Intern. Symposium on Neural Networks*, Chongqing, China, 2005, pp. 928-933.

[14] S. Keerthi, V. Sindhwani, and O. Chapelle, "An Efficient Method for Gradient-Based Adaptation of Hyperparameters in SVM Models," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. Cambridge: MIT Press, 2007, pp. 217-224.

[15] S. S. Keerthi and C.-J. Lin, "Asymptotic behaviors of support vector machines with Gaussian kernel," *Neural Computation*, 15(7), pp. 1667-1689, 2003.

[16] E. Kirkos, C. Spathis, and Y. Manolopoulos, "Data mining techniques for the detection of fraudulent financial statements," *Expert Systems with Applications*, 32(4), pp. 995-1003, 2007.

[17] P. Kumar, J. S. B. Mitchell, and E. A. Yildirim, "Approximate minimum enclosing balls in high dimensions using core-sets," *ACM Journal of Experimental Algorithmics*, 8, 2003. (http://doi.acm.org/10.1145/996546.996548).

[18] B. Lariviere and D. Van den Poel, "Predicting customer retention and profitability by using random forests and regression forests techniques," *Expert Systems with Applications*, 29(2), pp. 472-484, 2005.

[19] S. Lessmann and S. Voß, "A framework for customer-centric data mining with support vector machines," Working paper, *Institute of Information Systems, University of Hamburg*, 2007.

[20] J. C. Platt, "Fast Training of Support Vector Machines using Sequential Minimal Optimization," in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge: MIT Press, 1999, pp. 185-208.

[21] Prudsys, "The Data Mining Cup," 2007. (www.data-mining-cup.com)

[22] H. Shin and S. Cho, "Response modeling with support vector machines," *Expert Systems with Applications*, 30(4), pp. 746-760, 2006.

[23] L. C. Thomas, R. Oliver, and D. J. Hand, "A survey of the issues in consumer credit modelling research," *Journal of the Operational Research Society*, 56(9), pp. 1006-1015, 2005.

[24] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Core vector machines: Fast SVM training on very large data sets" *Journal of Machine Learning Research*, 6, pp. 363-392, 2005.

[25] I. W. Tsang, J. T. Kwok, and P.-M. Cheung, "Very Large SVM Training Using Core Vector Machines," *Proc. of the 10th Intern. Workshop on Artificial Intelligence and Statistics*, Barbados, 2005, pp. 349-356.

[26] I. W. Tsang, J. T. Kwok, and K. T. Lai, "Core Vector Regression for Very Large Regression Problems," *Proc. of the 22nd Intern. Conf. on Machine learning* Bonn, Germany, 2005, pp. 912-919.

[27] I. W. H. Tsang, J. T. Y. Kwok, and J. M. Zurada, "Generalized core vector machines," *IEEE Transactions on Neural Networks*, 17(5), pp. 1126-1140, 2006.

[28] T. Van Gestel, B. Baesens, J. A. K. Suykens, D. Van den Poel, D.-E. Baestaens, and M. Willekens, "Bayesian kernel based classification for financial distress detection," *European Journal of Operational Research*, 172(3), pp. 979-1003, 2006.

[29] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[30] S. Viaene, R. A. Derrig, B. Baesens, and G. Dedene, "A comparison of state-of-the-art classification techniques for expert automobile insurance claim fraud detection," *Journal of Risk & Insurance*, 69(3), pp. 373-421, 2002.

[31] J. H. Zar, *Biostatistical Analysis*, 4th ed. Upper Saddle River: Prentice Hall, 1999.