2006 International Joint Conference on Neural Networks
Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada
July 16-21, 2006

An Evaluation of Discrete Support Vector Machines for Cost-Sensitive Learning

Stefan Lessmann, Sven F. Crone, Robert Stahlbock, Nikolaus Zacher

Abstract— **The problem of cost-sensitive learning involves classification analysis in scenarios where different error types are associated with asymmetric misclassification costs. Business applications and problems of medical diagnosis are prominent examples and pattern recognitions techniques are routinely used to support decision making within these fields. In particular, support vector machines (SVMs) have been successfully applied, e.g. to evaluate customer credit worthiness in credit scoring or detect tumorous cells in bio-molecular data analysis. However, ordinary SVMs minimize a continuous approximation for the classification error giving similar importance to each error type. While several modifications have been proposed to make SVMs cost-sensitive the impact of the approximate error measurement is normally not considered. Recently, Orsenigo and Vercellis introduced a discrete SVM (DSVM) formulation [1] that minimize mis-classification errors directly and overcomes possible limitations of an error proxy. For example, DSVM facilitates explicit cost minimization so that this technique is a promising candidate for cost-sensitive learning. Consequently, we compare DSVM with a standard procedure for cost-sensitive SVMs and investigate to what extent improvements in terms of mis-classification costs are achievable. While the standard SVM performs remarkably well DSVM is found to give yet superior results.**

I. Introduction

THE support of decision making by means of classification analysis has received considerable attention in research and practice. Classification involves the prediction of a discrete class membership on the basis of observable/measurable attributes. For example, the task of credit scoring [2] consists of estimating whether a customer is credit worthy or not using attributes like the applicants age, income, occupation etc. It is generally believed that the costs of granting credit to a bad risk, e.g. a defaulting customer, is significantly greater than the cost of denying credit to a good risk candidate [3]. The same problem arises in medical diagnosis where a false alarm is usually not as severe as a missed correct alarm; e.g. missing a positive result when detecting tumors from magnetic resonance imaging scans might induce dramatic consequences while a small number of false alarms is tolerable if the scans are subsequently re-screened by medical personal; e.g. [4].

S. Lessmann (corresponding author), R. Stahlbock, N. Zacher, Institute of Information Systems, University of Hamburg, Von-Melle-Park 5, D-20146 Hamburg, Germany (phone: 0049-40-42838-5500; fax: 0049-40-42838-5535; e-mail: [lessmann, stahlbock]@econ.uni-hamburg.de, zacher.nikolaus@googlemail.com.

S. F. Crone, Department of Management Science, Lancaster University, Lancaster LA1 4YX, United Kingdom (e-mail: s.crone@lancaster.ac.uk).

The SVM, introduced by Vapnik an co-workers [5, 6], is a state-of-the-art classification algorithm that has been used successfully to support managerial and medical decision making; e.g. [7-10]. SVM training involves the minimization of a continuous approximation of the classification error giving similar importance to each error type. While the problem of using SVM for cost-sensitive classification has received some attention in the literature, [4, 11-14], the impact of the error approximation is usually not considered.

Recently, Orsenigo and Vercellis proposed a discrete SVM formulation that minimizes misclassification errors in a more intuitive manner. Emphasizing on classification errors and facilitating an explicit cost-minimization this approach is a promising candidate for cost-sensitive learning. Consequently, we compare DSVM with a standard procedure for cost-sensitive SVMs and evaluate to what extent improvements in terms of misclassification costs are achievable.

The remainder of the paper is organized as follows. A brief introduction to SVMs is given in Section II before we review previous work on cost-sensitive SVMs in Section III. We present the DSVM formulation in Section IV and describe a tabu search (TS) heuristic to solve the resulting optimization problem. The results of our empirical comparisons with the standard SVM can be found in Section V. Conclusions are given in Section VI.

II. Support Vector Machines

The original SVM can be characterized as a supervised learning algorithm capable of solving linear and non-linear binary classification problems. Given a training set with m patterns $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{m}$, where $\boldsymbol{x}_i \in X \subseteq \Re^n$ is an input vector and $y_i \in \{-1,+1\}$ its corresponding binary class label, the idea of support vector classification is to separate examples by means of a maximal margin hyperplane [15]. That is, the algorithm strives to maximize the distance between examples that are closest to the decision surface. It has been shown that maximizing the margin of separation improves the generalization ability of the resulting classifier [6]. To construct such a classifier one has to minimize the norm of the weight vector $\boldsymbol{w}$ under the constraint that the training patterns of each class reside on opposite sides of the separating surface; see Fig. 1. Since $y_i \in \{-1,+1\}$ we can formulate this constraint as:

$$y_i((\boldsymbol{w}\cdot \boldsymbol{x}_i)+b)\geq 1,\ i=1,...,m. \tag{1}$$

Examples which satisfy (1) with equality are called support vectors since they define the orientation of the resulting hyperplane.

To account for misclassifications, that is examples where constraint (1) is not met, the so called soft margin formulation of SVM introduces slack variables $\xi_i \in \Re$ [15]. Hence, to construct a maximal margin classifier one has to solve the convex quadratic programming problem (2):

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

β is a tuning parameter which allows the user to control the trade off between maximising the margin and classifying the training set without error. The primal decision variables $\boldsymbol{w}$ and b define the separating hyperplane, so that the resulting classifier takes the form:

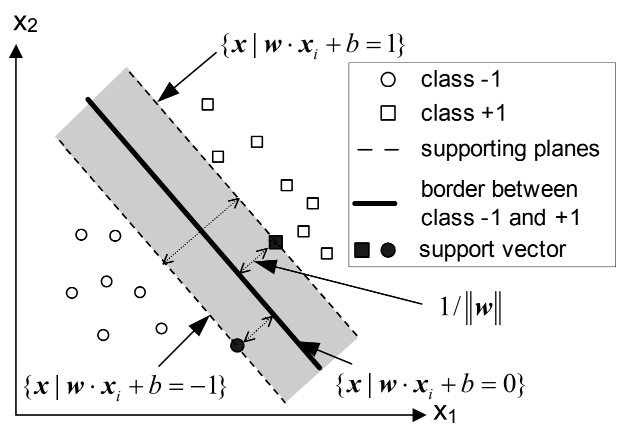
$$y(\boldsymbol{x}) = sgn\left((\boldsymbol{w}^*\cdot \boldsymbol{x})+b^*\right)\ , \tag{3}$$

where $\boldsymbol{w}^*$ and b^* are determined by (2).

To construct more general non-linear decision surfaces SVMs implement the idea to map the input vectors into a high-dimensional feature space via an a priori chosen non-linear mapping function Φ. Constructing a separating hyperplane in this feature space leads to a non-linear decision boundary in the input space. Expensive calculation of dot products $\Phi(\boldsymbol{x}_i)\cdot\Phi(\boldsymbol{x}_j)$ in a high-dimensional space can be avoided by introducing a kernel function K, see (4). The structure of SVMs allows this kernel integration without affecting the overall algorithms or training procedure [15].

$$K(\boldsymbol{x}_i,\boldsymbol{x}_j)=\Phi(\boldsymbol{x}_i)\cdot\Phi(\boldsymbol{x}_j). \tag{4}$$

Prominent candidates for the kernel function are the linear, radial and polynomial kernel; e.g. [15].



X2
$\{\boldsymbol{x} \mid \boldsymbol{w}\cdot \boldsymbol{x}_i + b = 1\}$
○ class -1
□ class +1
– – supporting planes
border between class -1 and +1
■ ● support vector
$1/\|\boldsymbol{w}\|$
$\{\boldsymbol{x} \mid \boldsymbol{w}\cdot \boldsymbol{x}_i + b = -1\}$
$\{\boldsymbol{x} \mid \boldsymbol{w}\cdot \boldsymbol{x}_i + b = 0\}$
X1

Fig. 1: Linear separation of two classes -1 and +1 in two-dimensional space with SVM classifier [16]

III. Approaches for Cost-Sensitive SVM

The problem of cost-sensitive learning is well established in the literature; e.g.[17, 18]. Let the entries c(i,j) of a loss-matrix $\boldsymbol{C}$ denote the cost of predicting class i when the true class is j. If we assume $\boldsymbol{C}$ to be asymmetric so that certain error types are more severe or costly than others, cost-sensitive learning refers to comprising this cost-information into the process of classifier induction. Approaches for cost-sensitive learning include algorithmic modifications to make individual learners cost-sensitive, e.g. [4, 14, 19] and meta-strategies designed to work with a broad variety of standard error based learners [20, 21]. Note that there is a strong connection between cost-sensitive learning and learning from imbalanced data sets so that these problems are commonly considered in a mutual framework [22, 23].

Following, we briefly review algorithmic modifications to make SVM cost-sensitive, and/or robust to skewed class distributions.

Considering the SVM classifier (2) and (3) there are three links to incorporate cost-sensitivity into SVM: The weight vector $\boldsymbol{w}$, the threshold b and the kernel K. The easiest way to bias SVM towards a minority and/or more important class is to manipulate the threshold b. This approach has been proposed by [14] and is also known as boundary movement [19] since the learned optimal separating hyperplane is altered as a post-processing step. The new resulting decision function is shown in (5) where the modifier Δb can be determined by ROC analysis [24].

$$y(\boldsymbol{x}) = sgn\left((\boldsymbol{w}^*\cdot \boldsymbol{x})+b^*+\Delta b\right) \tag{5}$$

A well established approach to consider imbalanced class/cost distributions during SVM learning is to modify the SVM objective using individual error weights for each class [4, 13].

$$\begin{aligned}&\min_{w,b,\xi} \frac{\beta}{2}\|\boldsymbol{w}\|+(1-\beta)\left(c^+\sum_{\{i|y_i=+1\}}\xi_i + c^-\sum_{\{i|y_i=-1\}}\xi_i\right)\\ &s.t.:\ y_i((\boldsymbol{w}\cdot \boldsymbol{x}_i)+b)\geq 1 -\xi_i,\ i=1,...,m.\end{aligned} \tag{6}$$

In (6), the weights c^+ and c^- measure the severity of a misclassification of positive and negative examples. It has been shown that this modification affects the weight vector $\boldsymbol{w}$ in the SVM decision function, e.g. [15]. We identify (6) to be the standard version of cost-sensitive SVM and will refer to it as csSVM in the following.

Adjustments of the kernel to overcome class imbalance problems have been suggested in [25, 26]. The authors propose a kernel boundary alignment algorithm that directly alters the kernel matrix increasing its values in the vicinity of the decision boundary and decreasing them in non boundary areas. This magnifies the spatial resolution of the training examples near the boundary, particularly in the area close to the minority class, and is shown to allow a purer data separation.

All approaches focus on effectiveness, e.g. classification accuracy, and do not explicitly consider costs. However, the most accurate classifier is not necessarily the most cost efficient one and to obtain a deeper integration of cost and decision making aspects, we propose a discrete SVM in the following.

IV. Discrete Support Vector Machines

A. Motivation and mathematical formulation

The original SVM utilizes the distance of a misclassified point to the separating hyperplane to measure classification error. That is, the discrete classification error is replaced by the continuous proxy ξ_i for computational convenience; see (2). DSVM [1] reverses this simplification and replaces ξ_i by $\theta_i \in [0,1]$ to account for asymmetric misclassification costs in a more intuitive manner. In addition, we substitute the L2-norm in SVMs' objective by the L1-norm. The L1-norm forces more elements of the weight vector to zero and therewith increases the interpretability of the model [27, 28]. Since model comprehensibility and transparency are deemed important in business and medical areas we believe the L1-norm to beneficial for these domains. Further more, using the L1-norm facilitates the usage of fast algorithms for solving linear programs as a sub-step within our tabu search heuristic; see IV.B for details.

The resulting formulation is given in (7) where u_j denotes a primal decision variable that controls the value of the weight vector $\boldsymbol{w}$ and Q is a sufficient large number.

$$\begin{aligned}
&\min_{w,b,\theta,u} \quad \frac{\beta}{2}\sum_{j=1}^{n} u_j + (1-\beta)\left(c^+ \sum_{\{i|y_i=+1\}} \theta_i + c^- \sum_{\{i|y_i=-1\}} \theta_i \right) \\
&s.t.: \quad y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b) \ge 1 - Q\theta_i, \quad i=1,...,m \\
&\qquad\quad -u_j \le w_j \le u_j \qquad j=1,...,n \\
&\qquad\quad \theta_j \in [0,1] \qquad j=1,...,n \\
&\qquad\quad u_j \ge 0 \qquad j=1,...,n.
\end{aligned} \tag{7}$$

Note that DSVM allows a different interpretation of c^+ and c^- than csSVM (6). From a decision making point of view these values serve as an abstract measure of error importance in csSVM that is used to weight a proxy for the classification error. As a rule of thumb the reciprocal of the class prior is a prominent choice for these parameters [27, 29]. On the contrary, in DSVM we can interpret c^+ and c^- a real cost values, e.g. in an economical sense. Consider for example the case of credit scoring. A false positive error, e.g. predicting a defaulting customer as credit worthy, is associated with a certain cost and while we could directly incorporate such values or respective estimates into (7) their usage in (6) is questionable due to the multiplication with a continuous distance. Hence, even if a user has an idea about class specific misclassification costs, there is no straightforward rule how to translate them into a suitable setting for c^+ and c^-. Furthermore, errors of the same type have in general varying impact on the objective depending on ξ_i. In fact, this is another reason why DSVM might be sounder for cost-sensitive learning.

Problem (7) is a linear program with continuous (w,b,u) and discrete (θ) decision variables.

As in the original SVM the objective includes a margin maximization part and an (empirical) error minimization part and therefore implement Vapnik's principle of structural risk minimization [6]. Additionally, we can interpret (7) as an approach to optimize generalization performance and misclassification cost in parallel.

Obviously, DSVM is computationally much more expensive than (6) due to the integer constraint. Since standard SVM optimization techniques are no longer applicable we develop a tabu search (TS) algorithm to be described in the following.

B. A tabu search heuristic for DSVM

To solve (7) we start with considering a relaxation of DSVM where the integer constraint for θ is dropped.

$$\begin{aligned}
&\min_{w,b,\theta,u} \quad \frac{\beta}{2}\sum_{j=1}^{n} u_j + (1-\beta)\left(c^+ \sum_{\{i|y_i=+1\}} \theta_i + c^- \sum_{\{i|y_i=-1\}} \theta_i \right) \\
&s.t.: \quad y_i((\boldsymbol{w}\cdot\boldsymbol{x}_i)+b) \ge 1 - Q\theta_i, \quad i=1,...,m \\
&\qquad\quad -u_j \le w_j \le u_j \qquad j=1,...,n \\
&\qquad\quad 0 \le \theta_j \le 1 \qquad j=1,...,n \\
&\qquad\quad u_j \ge 0 \qquad j=1,...,n
\end{aligned} \tag{8}$$

The linear program (8) provides an upper bound for the optimal solution of DSVM and can be solved efficiently using the simplex method; e.g. [30].

In order to solve the discrete SVM problem (7) we developed a tabu search (TS) algorithm. TS is a meta-heuristic to solve combinatorial optimization problems. The idea is to find a feasible solution and search its neighborhood for better candidates using local hill-climbing strategies. Here, better means higher/lower objective values for maximization/minimization problems. However, the TS objective does not necessarily coincide with the MIP's objective [1]. The name TS originates from the fact that the algorithms incorporates some heuristics which prohibit certain moves (tabu moves) to avoid cycling and stops at suboptimal points [31]; see [32-34] for details.

Our TS implementation is based on the observations that feasible solutions, and consequently also the optimal solution, for the zero-one problem (7) can be found in an extreme point of the relaxation (8); see e.g. [35]. Therefore, the extreme points of the polyhedral constraint region defined by (8) form a natural neighbourhood for TS and each extreme point is a basic feasible solution (BFS). The general structure of such an extreme point tabu search (EPTS) [35, 36] is as follows: 1) Use the simplex method to find an extreme point e for (8) and use it as an initial solution. 2) Examine adjacent solution in the neighbourhood of e. These are all solution that could be obtained by ordinary simplex pivot operations, e.g. exchanging a current basis variable for a non-basis variable. 3) Select the move that results in the largest improvement of the objective value and is not contained in the tabu list. 4) Execute the selected move and update the tabu list using information on the time a variable is pivoted (recency information) and its overall numbers of pivots (frequency information). To transform the generic EPTS schema in a concrete algorithm one has to define the strategy for screening the candidate list in step 2), the move evaluation function and the rules and memory structures for the tabu list.

Note that each TS move can increase/decrease the current objective value z and increase/decrease the current amount of integer infeasibility ii; that is the amount a given solution fails to fulfil the integer constraint. Therefore, every pivot operation belongs to one for four elementary types, e.g. increase z and decrease ii (best moves) or decrease z and increase ii (worst moves) [37].

Our TS implementation evaluates each move within the candidate list according to its move type. To resolve situations in which one can either increase z on the cost of increasing ii or decrease ii on the cost of decreasing z we incorporate a strategic oscillation component [35] so that the algorithms strives to improve z for a given number of iterations, then switches to decreasing ii, then switched back to z improvements, etc. This trade-off is illustrated in Fig. 2.

While the tabu status in our implementation is solely based on recency and frequency information (see above) we use an aspiration criterion to allow moves that lead to a new best feasible MIP solution even if they are currently tabu.

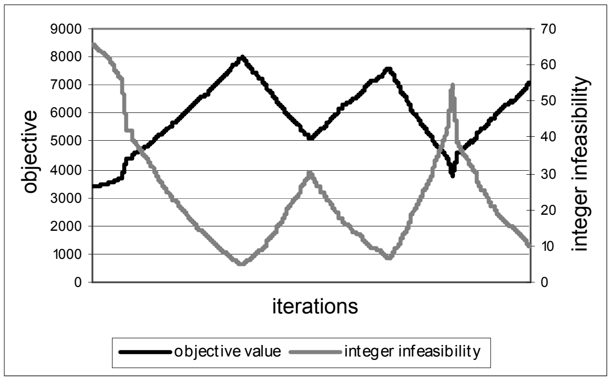


Fig. 2: Trade-off between TS moves that lower ii and improve objective value.

C. SVM decision tree

Introducing an integer constraint into the SVM formulations hinders application of the kernel trick to construct non-linear classifiers. In particular, the relaxed primal constraint $0 \le \theta_j \le 1$ in (8) results in a non-linear constraint in the dual problem. Consequently, solving the dual formulation is complicated. Note that this constraint is independent of the norm of the weight vector so that the same problem arises when using the L2-norm.

To overcome this limitation the construction of a hierarchical SVM decision tree has been proposed in [1]. We adopt this idea and recursively partition the data by linear DSVMs until some predefined conditions are met. To avoid over-fitting or the necessity to prune the SVM decision tree the following rules have to be regarded during the tree generation progress: 1) the tree deep, e.g. the number of levels, must not exceed a specified value, 2) splitting a node is impossible if the ratio of minority class examples falls below a specified threshold, and 3) any node must contain a specified minimum number of instances to remain divisible.

V. Empirical Study

A. Overview

The empirical evaluation of DSVM strives to explore the potential of DSVM and csSVM in cost-sensitive scenarios. We consider four data sets from the Statlog project [38] and the UCI machine learning library [39] as a case study. The two credit scoring data sets Australian credit (ac) and German credit (gc) serve as examples from the field of managerial decision making while heart-disease (hrt) and Wisconsin breast cancer (wbc) exemplify cases of medical diagnosis. A brief description of data set characteristics is given in Table 1. For detailed information on data set origin, task and variable description the reader is referred to [38, 40].

Table 1:
Data set characteristics*

	#Cases	#Features	#Class -1	#Class +1
ac	690	14	307	383
gc	1000	24	700	300
hrt	270	13	150	120
wbc	683	10	239	444

* We use the pre-processed data sets that are available via the LIBSVM homepage [41].

The data sets have been partitioned into 2/3 training set for model building and 1/3 test set for out-of-sample evaluation.

B. Predictive accuracy of DSVM

Preceding an analysis of asymmetric misclassification costs we have to verify the predictive accuracy of our DSVM implementation for standard situations. Therefore, we compare DSVM versus csSVM with linear (lin), radial (rad) and polynomial (poly) kernel function. Since DSVM is a linear classifier, we use the decision tree extension (Section IV.C) to construct hierarchical classifiers with a tree deep of two (DSVM-DT2) and three (DSVM-DT3) levels. The balanced error rate (BER) is used to measure predictive accuracy. BER is calculated as:

$$BER = 0.5\left(\frac{FP}{N_+} + \frac{FN}{N_-}\right). \quad (9)$$

Here, FP denotes the number of false positive cases, e.g. the number of false alarms while FN (false negatives) measures the number of missed true alarms. We use N_+ / N_- to represent the overall number of positive/negative examples with in data set.

Hyperparameters, e.g. β and kernel parameters have been determined by cross-validation (CV) adopting a grid search strategy [7, 42]. For each classifier, the parameter setting providing the lowest 10-fold CV BER is selected and evaluated on the test set. Results are given in Table 2.

TABLE 2:
BALANCED ERROR RATE OF CSSVM AND DSVM

	Training[1]			Test		
	BER	FP	FN	BER[2]	FP	FN
ac						
SVM (lin)	0.13	43	18	0.15	26	10
SVM (rad)	0.14	48	17	**0.14**	29	6
SVM (poly)	0.09	15	24	0.20	16	28
DSVM	0.13	46	15	0.19	31	8
DSVM-DT2	0.10	26	20	0.19	25	17
DSVM-DT3	0.09	26	15	0.19	28	14
gc						
SVM (lin)	0.32	46	98	0.34	13	66
SVM (rad)	0.32	30	63	0.34	17	63
SVM (poly)	0.32	30	72	0.32	14	61
DSVM	0.30	37	106	0.33	17	70
DSVM-DT2	0.28	37	106	**0.31**	14	65
DSVM-DT3	0.28	37	106	**0.31**	14	65
hrt						
SVM (lin)	0.14	12	14	0.21	6	10
SVM (rad)	0.15	11	17	0.22	4	12
SVM (poly)	0.13	10	14	**0.20**	5	10
DSVM	0.12	8	15	**0.20**	8	8
DSVM-DT2	0.08	8	7	0.21	7	10
DSVM-DT3	0.05	5	5	0.21	7	10
wbc						
SVM (lin)	0.03	5	8	0.04	3	3
SVM (rad)	0.03	4	9	**0.03**	2	5
SVM (poly)	0.02	4	4	0.07	7	4
DSVM	0.02	10	4	0.04	7	2
DSVM-DT2	0.02	5	5	0.04	4	3
DSVM-DT3	0.02	5	4	0.04	5	3

[1] Results on the training set are calculated by means of 10-fold CV.
[2] We use bold face to highlight the classifier that provides the lowest BER for the respective data set.

Table 2 reveals that DSVM's predictions are competitive to standard SVM for all data sets. In addition, the quality of DSVM is further confirmed by comparing our results with other benchmarking studies [1, 7, 43] that have used the same data. Having verified the predictive accuracy of our DSVM implementation we consider cost-sensitivity in the following section.

C. Cost-efficiency of DSVM

DSVM provides a more explicit integration of misclassification errors avoiding the use of a continuous error proxy. Therefore, we can expect DSVM to outperform standard SVM in scenarios involving asymmetric misclassification costs since it facilitates direct cost-minimization. In order to confirm this assumption we compare DSVM versus csSVM (6) under different cost-distributions contrasting their capability to derive cost-efficient predictions.

We consider applications where a false alarm is less severe than missing a correct one. Let C^+ denote the cost for a missed alarm, e.g. a bad credit risk, and C^- the costs for false alarm respectively. With no loss of generality we can set C^- to one and scale C^+ accordingly. Obviously, there is some point $\bar{C}^+ \gg C^-$ where a classifier completely avoids the more expensive error type FN. A pre-test revealed that this point is reached at latest at a ratio of $C^+ : C^- = 50:1$ for csSVM. Consequently our study incorporates cost distributions of $C^+ : C^- = 2:1$ to 50:1.

Aiming at a proximate translation of cost values into classifier parameters we considered a fixed setting for the c^+ and c^- parameters in csSVM and DSVM, see (6) and (7), directly using the respective $C^+ : C^-$ ratio. This decreases the number of free parameters and reflects the previously developed understanding of these parameters for DSVM. Subsequently, the remaining free parameter (kernel parameters and the trade-off parameter β) are determined by means of 10-fold CV using the resulting misclassification costs as selection criterion. That is, the classifier providing minimal misclassification costs within 10-fold CV is selected and evaluated on the test set. This procedure is repeated for each $C^+ : C^-$ ratio. Results are presented in Table 3.

TABLE 3:
RESULTS FOR CSSVM AND DSVM UNDER DIFFERENT COST-DISTRIBUTIONS WITH FIXED COST PARAMETERIZATION

	Results for csSVM				Results for DSVM			
Data set	Training		Test		Training		Test	
Cost ratio	FP	FN	FP	FN	FP	FN	FP	FN
ac								
2:1-5:1	40	10	32	7	46	15	31	8
6:1-10:1	50	10	36	6	49	11	34	7
11:1-15:1	100	0	57	3	102	6	58	3
16:1-20:1	120	0	68	3	141	3	74	3
25:1-50:1*	130	0	70	2	142	2	74	2
gc								
2:1-5:1	180	40	80	24	238	23	116	20
6:1-10:1	320	10	153	5	377	3	188	4
11:1-15:1	410	0	202	1	395	3	185	3
16:1-20:1	450	0	219	0	400	2	185	3
25:1-50:1*	450	0	221	0	400	2	185	3
hrt								
2:1-5:1	20	1	13	7	24	6	13	7
6:1-10:1	40	0	23	3	39	2	17	6
11:1-15:1	50	0	26	1	49	2	22	3
16:1-20:1	50	0	28	0	50	2	23	3
25:1-50:1*	60	0	28	0	63	1	30	1
wbc								
2:1-5:1	10	0	7	3	8	4	6	2
6:1-10:1	20	0	10	1	10	4	6	2
11:1-15:1	20	0	12	1	10	4	7	2
16:1-20:1	30	0	13	2	13	4	7	2
25:1-50:1*	40	0	15	1	22	2	11	1

* We used a step size of 5 to increase asymmetry in cost distributions from a ratio of 20:1 onwards. Consequently, this group contains approximately the same number of elements as the other groups.

We aggregate the different cost ratios into five groups and report the number of false positive and false negative predictions instead of one unique misclassification cost. Further more, csSVMs with linear, polynomial and radial kernel as well as DSVM with various tree levels are averaged for brevity of presentation.

The number of "expensive" false negative predictions is consistently decreased with increasing asymmetry of the $C^+ : C^-$ ratio for both classifiers. In fact, this type of error almost vanishes in highly asymmetric settings. Surprisingly, such results are not only obtained for DSVM but for csSVM as well. For the hrt data set, csSVM even outperforms DSVM on the test set in terms of FN errors. Regarding the less severe FP error type (false alarm) DSVM is slightly better than csSVM giving superior predictions in 13 out of 20 cases. While these results support our initial assumption that a direct integration of misclassification costs is suitable for DSVM, the competitive performance of csSVM has not been expected. On the one hand, this finding justifies the common application of csSVM for cost-sensitive learning. Further more, the overall low error rates actually motivate an adoption of our strategy for csSVM parameterization. The parameters c^+ and c^- are routinely set to the reciprocal of the prior probability for the positive / negative class [27, 29] and our results suggest that this rule of thumb can be extended to situations involving asymmetric misclassification costs.

In order to evaluate the implications of such a tuning heuristic, we conduct a second line of experiments allowing any possible setting for c^+ and c^- to obtain a cost minimal classifier for a given $C^+ : C^-$ ratio. That is, we allow usage of a csSVM or DSVM classifier with parameters settings of e.g. $c^+ = 2$ and $c^- = 1$ even if the assumed application domain involves a cost ratio of $C^+ : C^- = 50 : 1$. Consequently, the number of free parameters is increased dramatically. Results are given in Table 4.

Comparing Table 3 and Table 4 the larger number of free parameters has not helped to further improve csSVM results in general. We can observe a purer separation of the training data due to a larger number of degrees of freedom. Regarding to test set performance the number of FP errors has decreased at the cost of committing additional more severe FN errors. For the considered cost ratios this would induce overall higher misclassification costs. Consequently, the larger number of free parameters hinders effective model selection and leads to slightly deteriorate test results. This confirms that a proximate integration of misclassification costs is actually a good strategy for the considered data sets. In view of the fact, that there is to our best knowledge no broadly accepted procedure for SVM parameter setting in cost-sensitive scenarios, e.g. a cost-sensitive grid search, our results can be seen as a first step to develop a sophisticated tuning heuristic for csSVM.

TABLE 4:
RESULTS FOR CSSVM AND DSVM UNDER DIFFERENT COST-DISTRIBUTIONS WITH FREE COST PARAMETERIZATION

	Results for csSVM				Results for DSVM			
Data set	Training		Test		Training		Test	
Cost ratio	FP	FN	FP	FN	FP	FN	FP	FN
ac								
2:1-5:1	50	10	33	7	50	10	34	7
6:1-10:1	80	0	46	6	73	7	45	5
11:1-15:1	80	0	46	6	113	2	63	3
16:1-20:1	80	0	46	6	113	2	63	3
25:1-50:1	80	0	46	6	142	1	76	2
gc								
2:1-5:1	170	4	76	26	238	23	116	20
6:1-10:1	350	0	168	2	377	3	188	4
11:1-15:1	360	0	175	1	395	3	185	3
16:1-20:1	360	0	175	1	400	2	185	3
25:1-50:1	360	0	175	1	400	2	185	3
hrt								
2:1-5:1	20	0	12	7	27	4	13	7
6:1-10:1	20	0	13	6	34	2	16	6
11:1-15:1	20	0	13	6	45	1	21	3
16:1-20:1	20	0	13	6	60	0	28	0
25:1-50:1	20	0	13	6	60	0	28	0
wbc								
2:1-5:1	0	0	4	4	10	4	6	2
6:1-10:1	0	0	4	4	21	2	10	2
11:1-15:1	0	0	4	4	29	1	14	1
16:1-20:1	0	0	4	4	29	1	14	1
25:1-50:1	0	0	4	4	29	1	14	1

Results for the DSVM classifier have not changed significantly. This is explained by the fact that DSVM does not provide a large number of free parameters beside c^+ and c^-. While the novel setting allows a broader value range for these parameters, this flexibility is obviously not exploited so that the results remain stable. This verifies that the previous setting, e.g. direct translation of misclassification costs into parameter values, is indeed appropriate for DSVM and that further improvements are hardly achievable.

VI. CONCLUSIONS

We considered the case of cost-sensitive learning using SVM classifiers. DSVM appeared to be a very promising candidate for such scenarios due to its accurate error measurement. On the other hand, csSVM is the standard formulation for cost-sensitive classification with SVMs. However, the usage of a continuous approximation of misclassification error puts csSVMs appropriateness into perspective. Therefore, we compared csSVM and DSVM within an empirical experiment to evaluate their capabilities to derive cost-efficient predictions.

Our results confirmed that DSVM indeed provides highly accurate predictions when the distributions of misclassification error are uneven. In addition, we found csSVM to give surprisingly good results as well. In spite of its algorithmic treatment of classification errors, a direct translation of cost values into parameter settings emerged to be an efficient approach confirming and extending tuning heuristics for csSVM.

REFERENCES

[1] C. Orsenigo and C. Vercellis, "Discrete Support Vector Decision Trees via Tabu Search," *Computational Statistics & Data Analysis*, vol. 47, pp. 311-322, 2004.

[2] L. C. Thomas, "A Survey of Credit and Behavioral Scoring; Forecasting financial risk of lending to consumers," *International Journal of Forecasting*, vol. 16, pp. 149-172, 2000.

[3] D. West, "Neural network credit scoring models," *Computers & Operations Research*, vol. 27, pp. 1131-1152, 2000.

[4] Veropoulos, N. Cristianini, and C. Campbell, "Controlling the Sensitivity of Support Vector Machines," *Proc. of the 16th Intern. Joint Conf. on Artificial Intelligence*, Stockholm, Sweden, 1999.

[5] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," *Proc. of the 5th Annual Workshop on Computational Learning Theory*, Pittsburgh, Pennsylvania, USA, 1992.

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

[7] 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*, vol. 54, pp. 627-635, 2003.

[8] H. Shin and S. Cho, "Response modeling with support vector machines," *Expert Systems with Applications*, vol. In Press, Corrected Proof.

[9] Y. Wang, I. V. Tetko, M. A. Hall, E. Frank, A. Facius, K. F. X. Mayer, and H. W. Mewes, "Gene selection from microarray data for cancer classification - a machine learning approach," *Computational Biology and Chemistry*, vol. 29, pp. 37-46, 2005.

[10] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, "Gene Selection for Cancer Classification using Support Vector Machines," *Machine Learning*, vol. 46, pp. 389-422, 2002.

[11] G. Fumera and F. Roli, "Cost-sensitive learning in Support Vector Machines," *Proc. of the Workshop on Machine Learning, Methods and Applications*, Siena, Italy, 2002.

[12] P. Geibel, U. Brefeld, and F. Wysotzki, "Perceptron and SVM learning with generalized cost models," *Intelligent Data Analysis*, vol. 8, pp. 439-455, 2004.

[13] Y. Lin, Y. Lee, and G. Wahba, "Support Vector Machines for Classification in Nonstandard Situations," *Machine Learning*, vol. 46, pp. 191-202, 2002.

[14] G. Karakoulas and J. Shawe-Taylor, "Optimizing classifiers for imbalanced training sets," *Advances in Neural Information Processing Systems*, 1999.

[15] N. Cristianini and J. Shawe-Taylor, *An introduction to support vector machines and other kernel-based learning methods*. Cambridge: Cambridge University Press, 2000.

[16] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121-167, 1998.

[17] C. Elkan, "The Foundations of Cost-Sensitive Learning," *Proc. of the 7th Intern. Joint Conf. on Artificial Intelligence*, Seattle, Washington, USA, 2001.

[18] S. Viaene and G. Dedene, "Cost-sensitive learning and decision making revisited," *European Journal of Operational Research*, vol. 166, pp. 212-220, 2004.

[19] G. Wu and E. Y. Chang, "KBA: Kernel Boundary Alignment considering Imbalanced Data Distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 786-795, erscheint 2005.

[20] P. Domingos, "MetaCost: a general method for making classifiers cost-sensitive," *Proc. of the 5th Intern. Conf. on Knowledge Discovery and Data Mining*, San Diego, CA, USA, 1999.

[21] W. Fan, S. J. Stolfo, J. Zhang, and P. K. Chan, "AdaCost: Misclassification Cost-Sensitive Boosting," *Proc. of the 16th Intern. Conf. on Machine Learning*, Bled, Slovenia, 1999.

[22] N. Japkowicz and S. Stephen, "The Class Imbalance Problem: A Systematic Study," *Intelligent Data Analysis*, vol. 6, pp. 429-450, 2002.

[23] G. M. Weiss, "Mining with rarity: a unifying framework," *ACM SIGKDD Explorations Newsletter*, vol. 6, pp. 7-19, 2004.

[24] E. Carrizosa and B. Martin-Barragan, "Two-group classification via a biobjective margin maximization model," *European Journal of Operational Research*, appear 2006.

[25] G. Wu and E. Y. Chang, "Class-Boundary Alignment for Imbalanced Dataset Learning," in *ICML Workshop on Learning from Imbalanced Data Sets*. Washington DC, USA, 2003.

[26] G. Wu and E. Y. Chang, "KBA: Kernel Boundary Alignment considering Imbalanced Data Distribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 786-795, 2005.

[27] K. P. Bennett, S. Wu, and L. Auslender, "On support vector decision trees for database marketing," *Proc. of the Intern. Joint Conf. on Neural Networks*, Washington D.C., USA, 1999.

[28] P. S. Bradley and O. L. Mangasarian, "Feature Selection via Concave Minimization and Support Vector Machines," *Proc. of the 15th Intern. Conference on Machine Learning*, Madison, Wisconsin, 1998.

[29] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A Practical Guide to Support Vector Classification," Department of Computer Science and Information Engineering National Taiwan University, Taipei, Taiwan 2003.

[30] K. G. Murty, *Linear programming*. New York: Wiley, 1983.

[31] F. Glover, "Tabu search - wellsprings and challenges," *European Journal of Operational Research*, vol. 106, pp. 221-225, 1998.

[32] F. Glover and M. Laguna, *Tabu search*. Boston: Kluwer, 1997.

[33] F. Glover, "Tabu search: Part 1," *ORSA Journal on Computing*, vol. 1, pp. 190-206, 1989.

[34] F. Glover, "Tabu search: Part 2," *ORSA Journal on Computing*, vol. 2, pp. 4-32, 1990.

[35] A. Lokketangen and F. Glover, "Solving zero-one mixed integer programming problems using tabu search," *European Journal of Operational Research*, vol. 106, pp. 624, 1998.

[36] J. A. Blue and K. P. Bennett, "Hybrid extreme point tabu search," *European Journal of Operational Research*, vol. 106, pp. 676, 1998.

[37] A. Lokketangen and F. Glover, "Candidate List and Exploration Strategies for Solving 0/1 MIP Problems using a Pivot Neighborhood," in *Meta-Heuristics. Advances and Trends in Local Search Paradigms for Optimization*, S. Voß, S. Martello, I. H. Osman, and C. Roucairol, Eds. Boston: Kluwer, 1998.

[38] D. Michie, D. J. Spiegelhalter, and C. C. Taylor, *Machine learning, neural and statistical classification*. New York: Horwood, 1994.

[39] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz, "UCI Repository of machine learning databases," Department of Information and Computer Science, University of California, Irvine, CA, 1998.

[40] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih, "A Comparison of Prediction Accuracy, Complexity, and Training Time of Thirty-Three Old and New Classification Algorithms," *Machine Learning*, vol. 40, pp. 203-228, 2000.

[41] C.-C. Chang and C.-J. Lin, "LIBSVM - A Library for Support Vector Machines," Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

[42] T. van Gestel, J. A. K. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. de Moor, and J. Vandewalle, "Benchmarking Least Squares Support Vector Machine Classifiers," *Machine Learning*, vol. 54, pp. 5-32, 2004.

[43] Y. Lin, "Support Vector Machines and the Bayes Rule in Classification," *Data Mining and Knowledge Discovery*, vol. 6, pp. 259-275, 2002.