FORSCHUNGSZENTRUM JÜLICH GmbH
Zentralinstitut für Angewandte Mathematik
D-52425 Jülich, Tel. (02461) 61-6402

Technical Report

On the Advantages of Weighted L_1-Norm Support Vector Learning for Unbalanced Binary Classification Problems

*Tatjana Eitrich, Bruno Lang**

FZJ-ZAM-IB-2005-16

December 2005

(last change: 7.12.2005)

Preprint: submitted for publication

(*) Applied Computer Science and Scientific Computing Group,
Department of Mathematics, University of Wuppertal, Germany

On the Advantages of Weighted L_1-Norm Support Vector Learning for Unbalanced Binary Classification Problems

Tatjana Eitrich, Bruno Lang

Abstract— **In this paper we analyze support vector machine classification using the soft margin approach that allows for errors and margin violations during the training stage. Two models for learning the separating hyperplane do exist. We study the behavior of the resulting optimization algorithms in terms of training time and test accuracy for unbalanced data sets. The main goal of our work is to compare the features of the resulting classification functions, which are mainly defined by the support vectors arising during the support vector machine training.**

Index Terms— **Support Vector Machine Classification, Supervised Learning, Soft Margin Algorithms, Unbalanced Data.**

I. Introduction

During the last decade support vector machines (SVMs) have emerged as powerful and reliable kernel methods for binary classification tasks. Their application has been expanded to various fields of learning such as regression and clustering. In this work we analyze supervised SVM learning for two classes, which is one of the most important tasks of data mining in our days.

Usually the basic maximal margin classifier [1] is not well suited for learning real world problems. Either there is no solution at all or, when tuning the parameters, the hyperplane suffers from overfitting effects. To avoid such problems nearly all implementations use a soft margin model that includes a penalty parameter for the trade-off between training errors and model complexity.

In this work we analyze and compare the two well known SVM optimization approaches. We show that the L_1-norm learning method is superior to the L_2-norm method in terms of the number of support vectors in the training data. This method produces a significantly smaller number of support vectors, which results in a sparse classification function that leads to fast classification speed. In addition we show that using a special weighting method for the error penalization for unbalanced data the accuracy of this method is significantly better.

The remainder of this paper is structured as follows. In Section II we review some basic concepts of supervised learning and the support vector machine learning method. In Sect. III we introduce our flexible soft margin implementation that is used to run the tests. Results and a detailed discussion are given in Sect. IV. In Sect. V we summarize our findings and show directions to future work.

T. Eitrich is with the Central Institute for Applied Mathematics, Research Centre Juelich, Germany (e-mail: t.eitrich@fz-juelich.de).

B. Lang is with the Applied Computer Science and Scientific Computing Group, Department of Mathematics, University of Wuppertal, Germany (e-mail: lang@math.uni-wuppertal.de).

II. Soft Margin Support Vector Machine Learning Methods for Unbalanced Data

Support vector machine classification, as it was introduced by Vladimir Vapnik [2], is a well known machine learning method. We briefly review the basic facts that are important for the presentation of our work. For a detailed overview on SVM learning including the principle of kernel induced feature spaces and generalization theory we refer to [1], [3].

SVMs rely on a linear classification function

$$f_{\text{lin}}(\boldsymbol{x}) = \langle \boldsymbol{x}, \boldsymbol{w} \rangle_2 + b = \sum_{k=1}^{n} w_k x_k + b \quad (\boldsymbol{x} \in \mathbb{R}^n) \ . \tag{1}$$

With $n \in \mathbb{N}$ we denote the number of attributes. The values for the weight vector $\boldsymbol{w} \in \mathbb{R}^n$ and the threshold $b \in \mathbb{R}$ are fixed, but unknown and need to be adjusted during the so called SVM training on some data set. Afterwards binary classification for any $\boldsymbol{x} \in \mathbb{R}^n$ is achieved via a hypothesis function

$$h(\boldsymbol{x}) := \text{sgn}\left(f_{\text{lin}}(\boldsymbol{x})\right) \ , \tag{2}$$

where $\text{sgn}(\cdot)$ is the modified signum function, which we define as

$$\text{sgn(a)} = \begin{cases} 1 & \text{if} \ \ \text{a} \geq 0 \\ -1 & \text{else} \end{cases} \qquad (\text{a} \in \mathbb{R}) \ . \tag{3}$$

To extend the linear learning approach to a set of highly nonlinear classification functions the well known kernel trick [4] is applied. A function $\boldsymbol{\phi} : \mathbb{R}^n \rightarrow \mathcal{F}$ is used to map the data to a space $\mathcal{F}$ of possibly very high dimension $m \in \mathbb{N}$ to ensure linear separability of the data in the so called feature space $\mathcal{F}$. This leads to

$$f_{\text{nonlin}}(\boldsymbol{x}) = \langle \boldsymbol{\phi}(\boldsymbol{x}), \boldsymbol{w} \rangle_{\mathcal{F}} + b = \sum_{k=1}^{m} w_k \phi_k(\boldsymbol{x}) + b \ , \tag{4}$$

which is a function operating in the feature space. Given a data set of l training points (training set)

$$\left\{ (\boldsymbol{x}^i, y_i) \in \mathbb{R}^n \times \{-1, 1\}, \ 1 \leq i \leq l \right\} \ ,$$

support vector learning is based on the idea of maximizing the geometric margin between the two classes of points. As shown in Fig. 1, the margin is defined as the minimal distance between the training points and the separating hyperplane. Note that the hyperplane always lies in the middle of the empty region, thus the margin has equal values for both classes. Statistical learning theory provides

upper bounds for the generalization error and proves that the choice of the maximal margin hyperplane will lead to maximal generalization when predicting the classification of previously unseen examples.

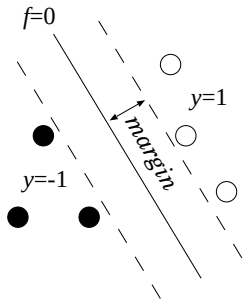

Fig. 1. Empty region with width of twice the margin size.

In the past the SVM learning method has been extended successfully to cope with noise in the training set. In the following we shortly describe the SVM soft margin approaches that are fundamental for real world data sets including noise. These models allow for margin errors and tend to produce robust classifiers that do not suffer from overfitting, which is an important objective of supervised learning.

The general SVM soft margin model simultaneously maximizes the width of the margin and minimizes the training errors [1]. A vector of slack variables $\boldsymbol{\xi}$ is used to measure by how much each training example fails to achieve a certain target margin γ in the feature space. For a training set the i-th slack variable $(1 \leq i \leq l)$ is defined as

$$\xi_i := \max \left\{ \gamma - y_i \left(\sum_{k=1}^{m} w_k \phi_k(\boldsymbol{x}^i) + b \right) ; 0 \right\} \in \mathbb{R} \ .$$

SVM training is aimed at minimizing the norm of the weight vector $\boldsymbol{w}$. This is due to the fact that SVMs have to maximize the geometric margin γ^g, which is defined as

$$\gamma^g := \frac{\gamma}{\|\boldsymbol{w}\|_{\mathcal{F}}} \ .$$

Therefore it is reasonable to fix the functional margin γ and minimize the norm of the weight vector. Traditionally $\gamma = 1$ is chosen.

The classification parameters of the nonlinear classifier (4) can be derived from the solution of the constrained convex optimization problem

$$\left.\begin{array}{ll} \min\limits_{\boldsymbol{w}\in\mathcal{F},\, b\in\mathbb{R},\, \boldsymbol{\xi}\in\mathbb{R}^l} & \frac{1}{2}\|\boldsymbol{w}\|_{\mathcal{F}}^2 + C\sum\limits_{i=1}^{l} \xi_i^q \\ \text{s.t.} & \\ & y_i \cdot f_{\text{nonlin}}(\boldsymbol{x}^i) \geq 1 - \xi_i \quad (1 \leq i \leq l) \ , \\ & \xi_i \geq 0 \qquad\qquad\quad (1 \leq i \leq l) \ . \end{array}\right\} \tag{5}$$

$C \geq 0$ is an important parameter for the compromise between margin and error. A small value for C will increase the number of training errors whereas a very large C will lead to a behavior similar to that of the hard margin SVM learning method that is not useful for most of the real world data sets. The overall margin violation is computed by summing up all values of the slack variables given the functional margin $\gamma = 1$ as a target [3].

The parameter $q \in \mathbb{N}_+$ is used to define the influence of the slack variables. SVM soft margin methods are divided into L_1 $(q = 1)$ and L_2 $(q = 2)$ models [1]. In Fig. 2 we show the consequence of this choice. For all $1 \leq i \leq l$ we have

- $\xi_i = 0$ for strong classifications outside the margin,
- $\xi_i \in (0, 1]$ for weak classifications inside the margin, and
- $\xi_i > 1$ for wrong classifications.

The L_2 model highly penalizes the real classification errors and tolerates margin violations to a greater extent, whereas the L_1 model handles both cases in a similar linear way.

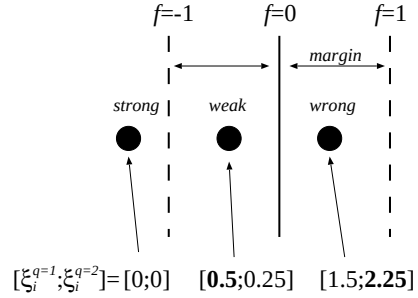

Fig. 2. Influence of the exponent q in (5) onto the error penalization for a negative training point. Results for positive points are evaluated analogously.

It is known that SVM algorithms solve the corresponding dual optimization problem to (5). This is mainly to allow for the usage of kernel functions. In the dual formulation only dot products $\langle \boldsymbol{\phi}(\boldsymbol{x}^i), \boldsymbol{\phi}(\boldsymbol{x}^j) \rangle$ between data points in the high-dimensional space $\mathcal{F}$ do occur; the points $\boldsymbol{\phi}(\boldsymbol{x}^i)$, $\boldsymbol{\phi}(\boldsymbol{x}^j)$ themselves are not required. Thus substituting the dot products with function values $k(\boldsymbol{x}^i, \boldsymbol{x}^j)$ of a nonlinear kernel function

$$k : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R} \tag{6}$$

relieves the user from constructing an explicit nonlinear mapping $\boldsymbol{\phi}$ for the input data.

For the primal problem (5) the dual problems can easily be derived by using Lagrange's theory [5]. The dual form for $q = 1$ is defined as

$$\left.\begin{array}{ll} \min\limits_{\boldsymbol{\alpha}\in\mathbb{R}^l} & \frac{1}{2}\sum\limits_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j k(\boldsymbol{x}^i, \boldsymbol{x}^j) - \sum\limits_{i=1}^{l} \alpha_i \\ \text{s.t.} & \\ & \sum\limits_{i=1}^{l} y_i \alpha_i = 0 \ , \\ & 0 \leq \alpha_i \leq C_i \qquad (1 \leq i \leq l) \ . \end{array}\right\} \tag{7}$$

Please note that we already included the kernel based formulation. In the same manner we derive the dual for

$q = 2$ as

$$\left.\begin{aligned} &\min_{\boldsymbol{\alpha}\in\mathbb{R}^l} \frac{1}{2}\sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j \left(k(\boldsymbol{x}^i,\boldsymbol{x}^j) + \frac{\delta_{ij}}{2C}\right) - \sum_{i=1}^{l}\alpha_i \\ &\text{s.t.} \\ &\qquad \sum_{i=1}^{l} y_i\alpha_i = 0 \ , \\ &\qquad 0 \le \alpha_i \qquad\qquad (1 \le i \le l) \ . \end{aligned}\right\} \quad (8)$$

The problem (7) differs from the maximal margin dual [1] in the Lagrange multipliers α_i $(1 \le i \le l)$ being upper bounded by C, which gives rise to call this approach the *box constraint*. The problem (8) is also very similar to the basic dual. The only change [6] is the addition of $1/2C$ to the diagonal entries in the Gram matrix $\boldsymbol{K}$ where

$$K_{i,j} := k(\boldsymbol{x}^i,\boldsymbol{x}^j) \quad (1 \le i,j \le l) \ .$$

Note that the usually the matrix $\boldsymbol{I}/C$ is added to $\boldsymbol{K}$ [7]. This corresponds to a slightly different formulation of (5), where a penalization constant $C/2$ is used instead of C. Sometimes even both approaches are presented in a mixed form [8], which may lead to problems in the comparison of tuned parameter values.[1] In this work we compare results of the soft margin methods for $q = 1$ and $q = 2$ and thus have to ensure usage of equal C values in (5).

Since we deal with convex problems the existence of unique global solutions for (7) and (8) is guaranteed [5]. Furthermore the optimal function values of primal and dual problems are equal and the primal solution vector is of the form

$$\boldsymbol{w}^* = \sum_{i=1}^{l} y_i \alpha_i^* \boldsymbol{\phi}(\boldsymbol{x}^i) \ . \quad (9)$$

Thus, for both methods the resulting dual solution $\boldsymbol{\alpha}^*$, i.e., the vector of Lagrange multipliers, can be used to define a dual classifier [1]

$$\begin{aligned} f^*_{\text{nonlin}}(\boldsymbol{x}) &\overset{(4)}{=} \sum_{k=1}^{m} w_k^* \phi_k(\boldsymbol{x}) + b^* \\ &\overset{(9)}{=} \sum_{k=1}^{m}\left(\sum_{i=1}^{l} y_i\alpha_i^*\phi_k(\boldsymbol{x}^i)\right)\phi_k(\boldsymbol{x}) + b^* \\ &= \sum_{i=1}^{l} y_i\alpha_i^* \langle \boldsymbol{\phi}(\boldsymbol{x}^i), \boldsymbol{\phi}(\boldsymbol{x})\rangle_{\mathcal{F}} + b^* \\ &\overset{(6)}{=} \sum_{i=1}^{l} y_i\alpha_i^* k(\boldsymbol{x}^i,\boldsymbol{x}) + b^* . \end{aligned}$$

Explicit knowledge of $\boldsymbol{w}^*$ is not required for applying (4). The threshold b^* can be determined using the so-called Karush–Kuhn–Tucker (KKT) conditions [1].

It is well known that only a small number of the training points is located within the margin or on the "wrong" side of the separating hyperplane. Only these points, called support vectors, have positive Lagrange multipliers and contribute to the final classifier

$$f^*_{\text{sparse}}(\boldsymbol{x}) = \sum_{\substack{1\le i\le l \\ \alpha_i > 0}} y_i\alpha_i^* k(\boldsymbol{x}^i,\boldsymbol{x}) + b^* \ . \quad (10)$$

In practice, however, for numerous problems the number of support vectors is very large. Sometimes this is caused by poorly tuned parameters. As a result the classifier (10) shows overfitting effects and is of slow classification speed.

In our work we analyze the impact of the parameter q onto this effect, especially for unbalanced classification problems. A data set is considered to be unbalanced if either the sizes of the two classes differ significantly, or the cost for a false negative classification is very high whereas a false positive is acceptable, or if both conditions hold. The latter is very important, e.g. for automated cost-sensitive cancer diagnosis [9] and the fast-check HIV-1/2 (serum) test [10]. For SVM learning typically a single penalization parameter is used for all training pairs i $(1 \le i \le l)$ [3]. It is reasonable to weigh wrong classifications of positive and negative points differently to obtain sensitive hyperplanes [11]. To this end we replace the single parameter C with two values [12] according to

$$C_i = \begin{cases} C^+ & \text{if } \ y_i = 1 \\ C^- & \text{otherwise} \end{cases} \quad (1 \le i \le l) \ . \quad (11)$$

As it is shown in Fig. 3 the choice of $C^+ > C^-$ induces a separating hyperplane which is much more distant from the smaller positive class than from the large negative one [8].

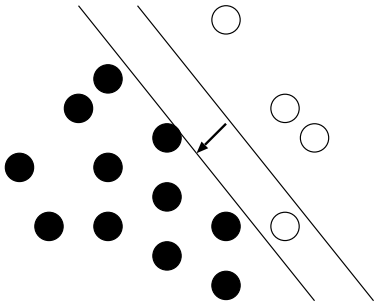

Fig. 3. Possibility of moving the separating hyperplane for unbalanced data.

III. Flexible L_1 and L_2 Implementation for Unbalanced Data

This work is based on the L_1-norm SVM training method described in [12], [13]. We briefly review the most important features and explain the adaption for unbalanced data as well as the flexible embedding of the L_2-norm learning.

We are working with the well known decomposition scheme [14] originally designed for the solution of (7). Modifications for the usage of the L_2-norm approach will be

[1] This problem also arises for the width of the Gaussian kernel, where sometimes the squared value is given.

explained later. The algorithm repeatedly performs the following four steps:
1. Select $\hat{l}$ "active" variables from the l free variables, the so-called working set. In our implementation the working set is made up from points violating the KKT conditions; see [12] for more details.
2. Restrict the optimization to the active variables and fix the remaining ones. Compute the kernel-submatrix $\boldsymbol{K}_{\text{active}} \in \mathbb{R}^{\hat{l}\times\hat{l}}$ for the restricted problem and the submatrix $\boldsymbol{K}_{\text{mixed}} \in \mathbb{R}^{(l-\hat{l})\times\hat{l}}$ for the stopping criterion.
3. Check for convergence. The solution is found if step 1. yields an empty working set.
4. Solve the restricted problem.

For each solution of the subproblem we use the generalized variable projection method described in [15]. It includes a fast inner solver given in [16]. The idea of splitting the quadratic problem into active and inactive parts iteratively is not new [17]. One feature that makes this approach particularly attractive for SVM training is the flexibility concerning the size $\hat{l}$. It can be chosen according to the available memory for the storage of the kernel matrices. Small values lead to a large number of fast iterations, whereas large values result in a small number of slow steps, since each solution of a subproblem means to solve a quadratic optimization problem with a dense matrix.

We adapted the software by replacing all occurrences of C with either C^+ or C^-, depending on the corresponding index i and the class label y_i. This mainly affects the working set selection routine, which is the famous method of Zoutendijk [18] and is usually applied for this task [19]. It can be shown that it corresponds to selecting points violating the KKT conditions [20], [21], [22] and thus is natural for SVM training. We described this modification in [12]. The other adjustments are for the data transformation in the inner solver and for the computation of the threshold b^* after the decomposition method solved the optimization problem. b^* is computed via

$$b^* := y_i - \sum_{\substack{1\le j\le l\\ \alpha_j^*>0}} y_j\alpha_j^* k(\boldsymbol{x}^j,\boldsymbol{x}^i) \qquad (12)$$

where we limit the possible l indices to

$$\left\{i : \alpha_i^* > 0,\ \alpha_i^* < C^+ \text{ for } y_i = 1,\ \alpha_i^* < C^- \text{ for } y_i = -1\right\}\ .$$

(12) results from the KKT conditions [1].

Based on the assumptions in Sect. II the modifications of the L_1-norm software for flexible usage of the L_2-norm approach are straight-forward to implement. The decomposition method remains unchanged, we only set $C^+ = C^- = \infty$. We tuned the kernel function according to (8) and the model (11). To this end we defined new parameters $\hat{C}^+$ and $\hat{C}^-$ that correspond to the original penalization values. The modified kernel is of the form

$$\hat{k}(\boldsymbol{x}^i,\boldsymbol{x}^j) := k(\boldsymbol{x}^i,\boldsymbol{x}^j) + \begin{cases} \frac{1}{2\hat{C}^+} & \text{if } \ i=j \ \text{ and } \ y_i = 1\\ \frac{1}{2\hat{C}^-} & \text{if } \ i=j \ \text{ and } \ y_i = -1\\ 0 & \text{else}\end{cases}$$

for all $1 \le i,j \le l$. It can be used for both methods. For the L_1-norm model it is important to set $\hat{C}^+ = \hat{C}^- = \infty$. The threshold for $q = 2$ is of the form

$$b^* := \begin{cases} \frac{\hat{C}^+-\alpha_i^*}{\hat{C}^+} - \sum\limits_{\substack{1\le j\le l,\\ 0<\alpha_j^*}} y_j\alpha_j^* K(\boldsymbol{x}^j,\boldsymbol{x}^i) & \text{if } \ y_i = 1\\ \frac{\alpha_i^*-\hat{C}^-}{\hat{C}^-} - \sum\limits_{\substack{1\le j\le l,\\ 0<\alpha_j^*}} y_j\alpha_j^* K(\boldsymbol{x}^j,\boldsymbol{x}^i) & \text{otherwise}\end{cases},$$

where i can be chosen among the indices of the support vectors in the training set [1].

IV. Experimental Evaluation and Interpretation

We analyzed two medical data sets, which are both publicly available from the UCI Machine Learning Repository [23].

The well-known breast cancer dataset from the University of Wisconsin Hospitals, Madison [24] includes 699 points, and each instance bears one of two possible class labels, benign or malignant. The number of malignant reference points is 241. From the ten attributes we removed the first one, since it codes the sample number and does not contribute relevant information. We defined the positive class to be malignant, so the dataset is slightly unbalanced and emphasis lies on high sensitivity. For obvious reasons we assume the cost for false negative points to be very high. Of the 699 points in the dataset, 349 were set aside for the final independent test. The remaining 350 points were used for the training. As shown in Tbl. I an adequate number of positive points is assigned to the test set. The percentage of training points is in fact small (50%), but it is large enough to compare results for different settings of the training methods.

TABLE I
Characteristics of the data sets and the class distributions.

	Training	Test
Cancer data set		
- Number of points	350	349
- Number of positive points	123	118
- Number of negative points	227	231
Thyroid data set		
- Number of points	2772	1000
- Number of positive points	190	94
- Number of negative points	2582	906

In addition we performed numerical experiments with the so-called thyroid data set from the Garavan Institute in Sydney, Australia, which is known to be a hard classification problem [25]. The 3772 instances have 15 binary and 6 continuous attributes. The task is to determine whether a patient is hypothyroid. The negative class represents 92% of the data [26]. Due to grossly unequal class sizes and high cost for false negative results, the data set is unbalanced. We used 2772 points for the training and the remaining

1000 points for the test. The class distributions are given in Tbl. I, too.

All in all we define 6 scenarios for our tests. Mainly we compare the behavior of the learning methods for $q = 1$ and $q = 2$, combined with the analysis of the influence by our approach (11), i.e.

- L_1-norm method $\begin{cases} C^+ = C^- \text{ with a large value (a)} \\ C^+ = C^- \text{ with a small value (b)} \\ C^+ \text{ and } C^- \text{ values differ (c)} , \end{cases}$
- L_2-norm method $\begin{cases} C^+ = C^- \text{ with a large value (d)} \\ C^+ = C^- \text{ with a small value (e)} \\ C^+ \text{ and } C^- \text{ values differ (f)} . \end{cases}$

TABLE II
TRAINING AND TEST RESULTS FOR THE CANCER DATA SET WITH THE L_1-NORM MODEL ($\sigma = 20.0$).

Case	(a)	(b)	(c)
Parameters $\{C^+, C^-\}$	$\{30, 30\}$	$\{2, 2\}$	$\{30, 2\}$
Training stage characteristics			
Number of support vectors	38	96	122
- positive	19	49	11
- negative	19	47	111
- free	5	5	12
- bounded	33	91	110
Number of training errors	11	10	10
- positive	4	4	**1**
- negative	7	6	**9**
Test stage results			
Number of test errors	11	15	8
- positive	6	9	**0**
- negative	5	6	**8**

For the training of the SVM with the decomposition method described in Sect. III we have chosen the following settings:

- There are 16 instances in the cancer data set that contain a single missing attribute value. We filled them with the mean values of the corresponding attributes.
- We scaled the data to zero mean and variance one.
- We selected the largest possible working set size $\hat{l} = l = 350$ for the cancer data set and an adequate working set size of $\hat{l} = l = 1000$ for the thyroid classification problem. Since we solve the quadratic programs (7) and (8), which have unique global solutions, the resulting classifiers do not depend on this internal parameter.
- The Gaussian kernel [4] was used for all tests. Based on results of optimization stages [12], [26] we fixed its width σ to 20 (cancer) and 100 (thyroid).
- For both data sets we define fixed parameter values for C^+ and C^-, which are based on earlier work [12], [26] and provide stable classifiers.

TABLE III
TRAINING AND TEST RESULTS FOR THE CANCER DATA SET WITH THE L_2-NORM MODEL ($\sigma = 20.0$).

Case	(d)	(e)	(f)
Parameters $\{C^+, C^-\}$	$\{30, 30\}$	$\{2, 2\}$	$\{30, 2\}$
Training stage characteristics			
Number of support vectors	95	196	227
- positive	45	67	15
- negative	50	129	212
- free	70	165	187
- bounded	25	31	40
Number of training errors	11	10	19
- positive	4	4	**0**
- negative	5	6	**19**
Test stage results			
Number of test errors	12	15	21
- positive	7	10	**0**
- negative	5	5	**21**

In Tbl. II and Tbl. III our results are presented for the training stages as well as for the tests with the breast cancer data set. We show the number of support vectors in the training set as well as their distribution among the positive and negative classes. In addition we distinguish between the free support vectors with $\alpha_i \in (0, C_i)$ and the bounded support vectors with $\alpha_i \equiv C_i$. The weighted model with $q = 1$ performed best on the test data, the sensitivity was 100% and the test results were similar to the training results, which indicates good generalization of the model. The L_2 model seems to produce too many support vectors. For all tests their number was twice as high compared to $q = 1$. Interestingly, both methods showed the same behavior for the unweighted tests, i.e. the first two columns in the tables. By contrast, the L_1 model was superior to L_2 when using the weighted approach. Both methods achieved high sensitivity, but the number of false positive points was significantly lower for $q = 1$.

Tables IV and V give the corresponding data for the thyroid disease data set. These results indicate an even more pronounced superiority of training the L_1-norm model with weighted error parameters for this large and highly unbalanced data set. We summarize

- The number of support vectors in the training data differ by a factor larger than 4 where the L_1-norm model was always superior.
- The free and bounded support vectors show a reverse distribution for the L_2-norm model.
- Usage of a weighted model always led to a reverse behavior of sensitivity and specificity for the training as well as for the test points.
- The best test result in terms of accuracy and sensitiv-

TABLE IV

Training and test results for the thyroid data set with the L_1-norm model ($\sigma = 100.0$).

Case	(a)	(b)	(c)
Parameters $\{C^+, C^-\}$ (divided by 10^3)	{100, 100}	{10, 10}	{100, 10}
Training stage characteristics			
Number of support vectors	189	217	323
- positive	87	104	36
- negative	102	113	287
- free	35	21	23
- bounded	155	196	300
Number of training errors	44	64	64
- positive	37	57	**1**
- negative	7	7	**63**
Test stage results			
Number of test errors	27	33	25
- positive	25	31	**1**
- negative	2	2	**24**

TABLE V

Training and test results for the thyroid data set with the L_2-norm model ($\sigma = 100.0$).

Case	(d)	(e)	(f)
Parameters $\{C^+, C^-\}$ (divided by 10^3)	{100, 100}	{10, 10}	{100, 10}
Training stage characteristics			
Number of support vectors	790	1096	1584
- positive	118	126	94
- negative	672	970	1490
- free	661	964	1250
- bounded	129	132	334
Number of training errors	68	93	126
- positive	63	88	**2**
- negative	5	5	**124**
Test stage results			
Number of test errors	40	48	45
- positive	37	47	**1**
- negative	3	1	**44**

ity was achieved with the weighted L_1-norm model. The number of errors in the L_2-norm model was twice as large. Tests with various real world data sets from pharmaceutical industry [27], [28] yielded similar results and led to the improvement of our data mining pipeline.

TABLE VI

Training times (in seconds) for the six settings.

Case	(a)	(b)	(c)	(d)	(e)	(f)
Cancer data	0.07	0.07	0.11	0.09	0.05	0.09
Thyroid data	422	141	203	150	48	98

We complete the discussion by giving the training times for our tests in Tbl. VI. Besides classification speed, the amount of time for SVM training plays a crucial role for the selection of algorithms. As it is shown the model (c) with the best performance led to the most expensive SVM training for the cancer data and to a very expensive training for the thyroid data. However, for unbalanced classifications problems where emphasis lies on true positive points and where a false negative point produces high costs, the observed increase in training time seems to be acceptable. For large-scale learning problems where a single training stage is extremely expensive we propose usage of parallel SVM learning [29], [30], [31].

V. Concluding Remarks and Future Work

We have presented a comparison of the two most frequently used SVM training methods for the classification of unbalanced data sets where high emphasis is put on sensitivity. Experiments have shown that the L_1-norm model in conjunction with a flexible error weighting is adequate to achieve high sensitivity in the training and test data and furthermore minimizes the number of false positive classifications.

Future research directions include kernel modifications in conjunction with parameter optimization for unbalanced data sets.

Acknowledgements

We greatly acknowledge the support by Research Centre Juelich.

References

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

[2] V. N. Vapnik, *Statistical learning theory*, John Wiley & Sons, New York, 1998.

[3] B. Schölkopf and A. J. Smola, *Learning with kernels*, MIT Press, Cambridge, MA, 2002.

[4] O. Chapelle, V. N. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 131–159, 2002.

[5] R. Fletcher, *Practical methods of optimization, Vol II: constrained optimization*, John Wiley & Sons, Chichester and New York, 1981.

[6] K.-M. Lin and C.-J. Lin, "A study on reduced support vector machines," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1449–1559, 2003.

[7] G. Lanckriet T. De Bie and N. Cristianini, "Convex tuning of the soft margin parameter," Tech. Rep. UCB/CSD-03-1289, EECS Department, University of California, Berkeley, 2003.

[8] F. Markowetz, "Support vector machines in bioinformatics," M.S. thesis, University of Heidelberg, 2001.

[9] S. Merler, C. Furlanello, B. Larcher, and A. Sboner, "Automatic model selection in cost-sensitive boosting," *Information Fusion*, vol. 4, no. 1, pp. 3–10, 2003.

[10] M. L. Rekart, M. Krajden, D. Cook, G. McNabb, T. Rees, and J. Isaac-Renton et al., "Problems with the fast-check hiv rapid test kits," *Canadian Medical Association Journal CMAJ*, vol. 167, no. 2, 2002.

[11] J. Drish, "Obtaining calibrated probability estimates from support vector machines," 2001, http://www-cse.ucsd.edu/users/jdrish/svm.pdf.

[12] T. Eitrich and B. Lang, "Efficient optimization of support vector machine learning parameters for unbalanced datasets," *Journal of Computational and Applied Mathematics*, (in press).

[13] T. Eitrich and B. Lang, "Efficient implementation of serial and parallel support vector machine training with a multi-parameter kernel for large-scale data mining," Preprint FZJ-ZAM-IB-2005-12, Research Centre Juelich, October 2005, submitted for publication.

[14] P. Laskov, "Feasible direction decomposition algorithms for training support vector machines," *Machine Learning*, vol. 46, no. 1-3, pp. 315–349, 2002.

[15] T. Serafini, G. Zanghirati, and L. Zanni, "Gradient projection methods for quadratic programs and applications in training support vector machines," *Optimization Methods and Software*, vol. 20, no. 2-3, pp. 353–378, 2005.

[16] P. M. Pardalos and N. Kovoor, "An algorithm for a singly constrained class of quadratic programs subject to upper and lower bounds," *Mathematical Programming*, vol. 46, no. 3, pp. 321–328, 1990.

[17] S. Leyffer, "The return of the active set method," 2005, to appear in Oberwolfach Report.

[18] G. Zoutendijk, *Methods of feasible directions: a study in linear and non-linear programming*, Elsevier, 1960.

[19] T. Joachims, "Making large-scale SVM learning practical," in *Advances in Kernel Methods – Support Vector Learning*, Bernhard Schölkopf, Christopher Burges, and Alexander Smola, Eds., pp. 169–185. MIT Press, 1998.

[20] C.-J. Lin, "Linear convergence of a decomposition method for support vector machines," 2001, http://www.csie.ntu.edu.tw/∼cjlin/papers/linearconv.pdf.

[21] C.-J. Lin, "On the convergence of the decomposition method for support vector machines," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1288–1298, 2001.

[22] C.-C. Chang, C.-W. Hsu, and C.-J. Lin, "The analysis of decomposition methods for support vector machines," *IEEE Transactions on Neural Networks*, vol. 11, no. 4, pp. 1003–1008, July 2000.

[23] S. Hettich, C. L. Blake, and C. J. Merz, *UCI Repository of machine learning databases*, 1998, http://www.ics.uci.edu/∼mlearn/MLRepository.html.

[24] O. L. Mangasarian and W. H. Wolberg, "Cancer diagnosis via linear programming," *SIAM News*, vol. 23, pp. 1–18, 1990.

[25] J. R. Quinlan, "Simplifying decision trees," in *Knowledge Acquisition for Knowledge-Based Systems*, B. Gaines and J. Boose, Eds., pp. 239–252. Academic Press, London, 1988.

[26] T. Eitrich and B. Lang, "Parallel tuning of support vector machine learning parameters for large and unbalanced data sets," in *Computational Life Sciences, First International Symposium, CompLife 2005, Konstanz, Germany, September 25-27, 2005, Proceedings*, M. R. Berthold, R. Glen, K. Diederichs, O. Kohlbacher, and I. Fischer, Eds. 2005, vol. 3695 of *Lecture Notes in Computer Science*, pp. 253–264, Springer.

[27] A. Kless and T. Eitrich, "Cytochrome p450 classification of drugs with support vector machines implementing the nearest point algorithm," in *Knowledge Exploration in Life Science Informatics, International Symposium, KELSI 2004, Milan, Italy, November 25-26, 2004, Proceedings*, Jesús A. López, Emilio Benfenati, and Werner Dubitzky, Eds. 2004, vol. 3303 of *Lecture Notes in Computer Science*, pp. 191–205, Springer.

[28] T. Eitrich and B. Lang, "Analysis of support vector machine training costs for large and unbalanced data from pharmaceutical industry," Preprint FZJ-ZAM-IB-2005-07, Research Centre Juelich, June 2005, submitted for publication.

[29] G. Zanghirati and L. Zanni, "A parallel solver for large quadratic programs in training support vector machines," *Parallel Computing*, vol. 29, no. 4, pp. 535–551, 2003.

[30] T. Serafini, G. Zanghirati, and L. Zanni, "Parallel decomposition approaches for training support vector machines," in *Proceedings of the International Conference ParCo2003, Dresden, Germany*. 2004, Hardbound.

[31] T. Eitrich and B. Lang, "Shared memory parallel support vector machine learning," Preprint FZJ-ZAM-IB-2005-11, Research Centre Juelich, September 2005, submitted for publication.