Contents

Sphered Support Vector Machine

Sepp Hochreiter, Johannes Mohr, and Klaus Obermayer
Department of Electrical Engineering and Computer Science
Technische Universität Berlin
10587 Berlin, Germany
{hochreit,johann,oby}@cs.tu-berlin.de

Abstract

We introduce an objective function for support vector machines (SVMs) which is scale invariant and leads to improved bounds on the generalization error. Standard SVMs and their margin bounds are not invariant under linear transformation. The new objective function leads to new SVM approach, called "Sphered Support Vector Machine" (S-SVM). The S-SVMs are kernelized and regularized but can be fast implemented by an incremental learning method and do not require kernel PCA like the approaches in (Chapelle and Schölkopf, 2002). On real world benchmark datasets we compare the new S-SVMs to standard ν-SVMs, where the S-SVMs showed comparable to superior classification performance to ν-SVMs.

1 Introduction

Support vector machines (SVMs) (Boser et al., 1992; Vapnik, 1995, 1998; Schölkopf and Smola, 2002) are now well established in the machine learning community and showed success in various applications. However the selected classifiers as well as the bounds on the generalization error depend on the data preprocessing even if it is linear. How should the data normalized for SVMs to obtain good classification performance? A general rule for data preprocessing is missing for SVMs.

For artificial neural networks it is common knowledge that normalizing the input variables to zero mean and unit variance has advantages for learning, e.g. (Bishop, 1995) Section 8.2. That normalization can be of advantage also for SVMS was shown in (Vapnik, 1995) for handwritten digit recognition. In (Herbrich and Graepel, 2000) the conventional margin bound was improved through the normalized margin, i.e. if the margin is divided by the length of the data vector producing the margin. As a consequence in (Herbrich and Graepel, 2000) the authors recommend "When training an SVM, always normalize your data in feature space". This statement was further confirmed by experiments in (Herbrich and Graepel, 2000). Normalizing the data in feature space is straightforward through the kernel matrix. However, here we want to go one step further and sphere the data in feature space.

Sphering the data in features space is implicitly introduced by a new objective. It results from a new generalization error bound which simultaneously estimates the misclassification error and the function class. Previous bounds (Vapnik, 1995; Schölkopf and Smola, 2002) assume a function class by assuming an input domain which is based on the observed training data. But it is not guaranteed that a new data point does not exceed the boundaries imposed by the training data. The new objective can also be derived by the framework of (Schölkopf et al., 1998; Chapelle and Schölkopf, 2002) by requiring scale invariance.

We base our approach on new bounds and go beyond the framework of (Schölkopf et al., 1998; Chapelle and Schölkopf, 2002) at two aspects. First, we derive for the linear case techniques which are also applicable if the quadratic part of the objective function is singular. Secondly, the kernelized version does not rely on kernel PCA because we use special properties of the scaling invariance.

2 Sphered Support Vector Machine

2.1 New Machine

We consider a classification problem, where every object is described by feature vector $\boldsymbol{x} \in \mathbb{R}^N$ belongs to one of two classes. A classifier is selected based on L objects $\boldsymbol{x}^i$, $1 \leq i \leq L$ and their binary labels $y_i \in \{+1, -1\}$. This training set $X = \left\{\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^L\right\}$ is summarized by a data matrix $\boldsymbol{X} := \left(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^L\right) \in \mathbb{R}^{N \times L}$ and a label vector $\boldsymbol{y} := (y_1, y_2, \ldots, y_L)$. The classifier is selected from the set $\{\text{sign}\,(f)\}$ of linear classifiers with

$$\text{sign}\,(f) \;=\; \{(\boldsymbol{x}, y) \mid \; y \;=\; \text{sign}\,(f(\boldsymbol{x})) \;=\; \text{sign}\,(\langle \boldsymbol{w}, \boldsymbol{x} \rangle \;+\; b)\} \tag{1}$$

which are parameterized by the weight vector $\boldsymbol{w}$ and the offset b, where $\langle \cdot, \cdot \rangle$ denotes a dot product. We assume that the parameter $\boldsymbol{w}$ and b of the classifiers are scaled, such that the hyperplane $f(\boldsymbol{x}) \;=\; 0$ is in its "canonical form" (Vapnik, 1995), i.e. $\min_{\boldsymbol{x} \in X} |\langle \boldsymbol{w}, \boldsymbol{x} \rangle \;+\; b| \;=\; 1$ holds.

The "margin" γ is the distance of the hyperplane to the closest data point

$$\gamma \;=\; \frac{\min_{\boldsymbol{x} \in X} |\langle \boldsymbol{w}, \boldsymbol{x} \rangle \;+\; b|}{\|\boldsymbol{w}\|_2} \;. \tag{2}$$

and can be expressed for hyperplanes in the canonical form by $\gamma \;=\; \|\boldsymbol{w}\|_2^{-1}$. Conventional support vector machines (SVMs, Vapnik, 1998; Schölkopf and Smola, 2002) select the hyperplane in canonical form which correctly classifies the training data and maximizes the margin:

$$\begin{aligned} \min_{\boldsymbol{w}, b} \quad & \frac{1}{2}\, \|\boldsymbol{w}\|^2 \qquad\qquad (3) \\ \text{s.t.} \quad & y_i \;\left(\left\langle \boldsymbol{w}, \boldsymbol{x}^i \right\rangle \;+\; b\right) \;\geq\; 1 \;\;. \end{aligned}$$

To maximize the margin was motivated by bounds on the generalization error using the Vapnik-Chervonenkis (VC) dimension h as capacity measure (e.g. Vapnik, 1998). For linear classifiers on X with $\gamma \geq \gamma_{\min}$ the VC dimension can be bounded by

$$h \ \leq \ \min \left\{ \left[\frac{\mathrm{R}^2}{\gamma_{\min}^2} \right] \ , \ N \right\} \ + \ 1 \tag{4}$$

(see Vapnik, 1998; Schölkopf and Smola, 2002). $[\cdot]$ denotes the integer part, and R is the radius of the smallest sphere in data space, which contains all the training data. Similar bounds which depend on $\frac{\mathrm{R}}{\gamma_{\min}}$ can be derived for other capacity measures e.g. for the fat shattering dimension (Shawe-Taylor et al., 1996, 1998; Schölkopf and Smola, 2002).

If the sphere containing all data is centered at the origin and $\hat{\boldsymbol{w}} \ := \ \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$ is an unit vector, then

$$\mathrm{R} \ = \ \max_{i, \hat{\boldsymbol{w}}} \left| \langle \hat{\boldsymbol{w}}, \boldsymbol{x}^i \rangle \right| \ = \ \max_{\hat{\boldsymbol{w}}} m(f, \boldsymbol{X}) \ , \tag{5}$$

where

$$m(f, \boldsymbol{X}) \ := \max_i \left| \langle \hat{\boldsymbol{w}}, \boldsymbol{x}^i \rangle \right| \ . \tag{6}$$

Because $m(f, \boldsymbol{X})$ depends on $\boldsymbol{w}$, as γ does, it seems reasonable to include $m(f, \boldsymbol{X})$ in a capacity measure instead of R. Indeed, in Section 4 we show that $\left(\frac{m(f, \boldsymbol{X})}{\gamma} \right)^2$ is an upper bound on the generalization error which extends the known bounds in the SVM context (Vapnik, 1998; Schölkopf and Smola, 2002) (see Section 4 for more details) .

Another disadvantage of using R in the cpacity measure and, therefore, only maximizing the margin by the SVM thechnique is that both the bound and the classifier selection are not invariant under linear transformations of the data. That means preprocessing can considerably change the result. But what is the best preprocessing? For the bound this invariance can be easily seen. If we scale with factor s all directions orthogonal to the normal vector of the hyperplane $\boldsymbol{w}$, then the margin does not change. However, R maybe scaled, whereas $m(f, \boldsymbol{X})$ is not scaled. R is scaled to a minimal value, if $s = 0$. This situation is depicte in Fig. 1. That the selected classifier also changes with scaling, is shown in in the left hand side panel of Fig. 2. The figure shows the separating hyperplane (top), the separating hyperplane after scaling the data in one direction (middle), and both hyperplanes in the orginal data where the separating hyperplane after scaling is projected back. It can be seen that the hyperplanes differ from one another and, therefore, new data points can be classified differently by both classifiers – also the number of support vectors changed.

We now upper bound the scale invariant capacity measure $\frac{m(f, \boldsymbol{X})}{\gamma}$:

$$\frac{m(f, \boldsymbol{X})}{\gamma} \ = \ \|\boldsymbol{w}\|_2 \ \max_i \left| \langle \hat{\boldsymbol{w}}, \boldsymbol{x}^i \rangle \right| \ \leq \ \sqrt{\sum_i \left(\langle \boldsymbol{w}, \boldsymbol{x}^i \rangle \right)^2} \ = \ \|\boldsymbol{X}^T \ \boldsymbol{w}\|_2 \tag{7}$$

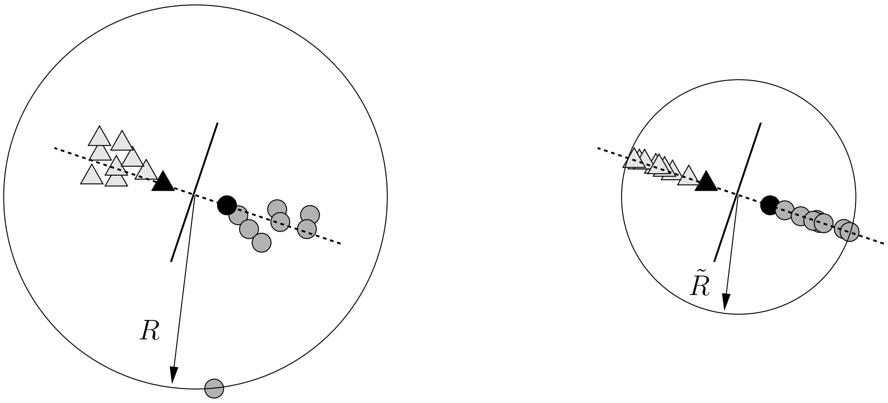

Figure 1: Left: The orginal data belonging to one of two classes (indicated by circles and triangles) and the separating hyperplane with maximal margin (the closest point to the hyperplane are marked black). Right: The same data but all directions orthogonal to the normal vector of the separating hyperplane are scaled to zero. The radius R of the sphere containing all data is scaled to a smaller radius R̃ whereas the margin remained constant.

and choose as new objective function

$$\|\boldsymbol{X}^T \; \boldsymbol{w}\|_2^2 \; . \tag{8}$$

As will be shown below on page 7, the new objective function is invariant under linear transformation with full rank.

Our objective function is related to approaches which are designed to handle invariances. In (Schölkopf et al., 1998), and in (Chapelle and Schölkopf, 2002) for the nonlinear case, the classifier should be invariant against local transformations $\mathcal{L}_t$ of the data vectors. A scaling invariance $\mathcal{L}_t(\boldsymbol{x}) = (1+t)\boldsymbol{x}$ leads to the covariance matrix of tangent vectors $C = \boldsymbol{X}\boldsymbol{X}^T$, i.e. our new objective through $\boldsymbol{w}^T \; C \; \boldsymbol{w}$. But there is a big difference between our approach and the approach in (Schölkopf et al., 1998) and in (Chapelle and Schölkopf, 2002). We use the $\|\boldsymbol{X}^T \; \boldsymbol{w}\|_2^2$ as a term measuring the capacity or the complexity of a classifier which must not be zero. In (Schölkopf et al., 1998) and in (Chapelle and Schölkopf, 2002) the term should be zero in order to enforce the according invariances.

Now we derive the new method for model selection based on the new objective function. The constraints for correct classification $y^i \; f(\boldsymbol{x}^i) \geq 1$ on the training set $\boldsymbol{X}$ together with the canonical form of the hyperplane are enforced through

$$1 \; \leq \; y^i \; \left(\langle \boldsymbol{w}, \boldsymbol{x}^i \rangle \; + \; b \right) \; . \tag{9}$$

Now we obtain the **Sphered Support Vector Machine (S-SVM)** optimiza-

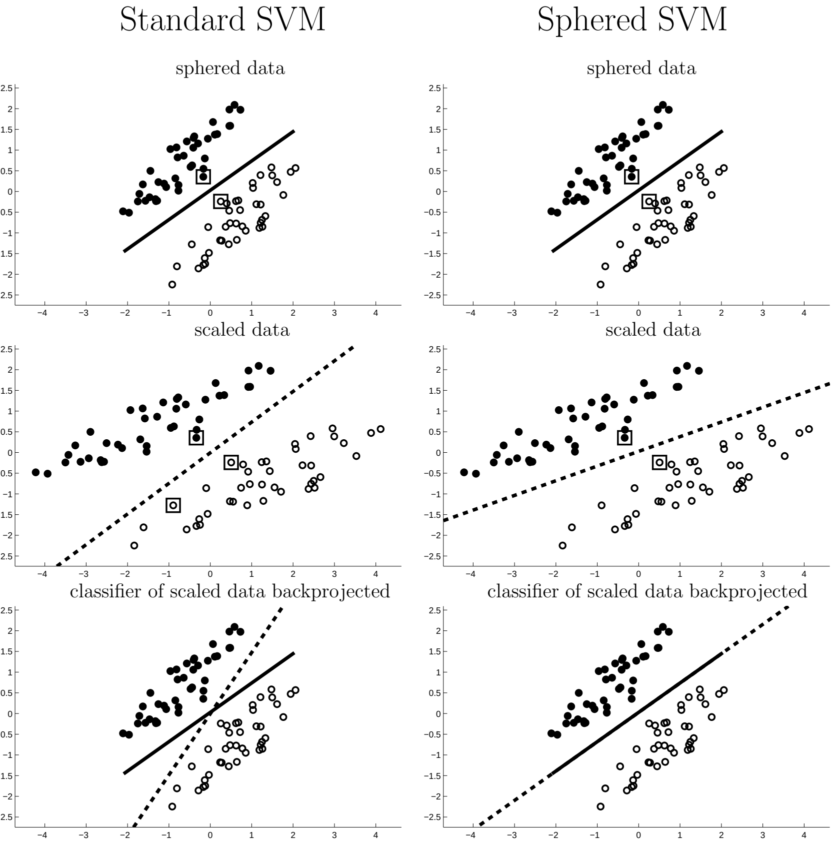

Figure 2: The sphered SVM (right) and the standard SVM (left) on a simple data set after and before scaling. If the sphered SVM solution on the scaled data is scaled back, then it coincindeces with the orginal solution. In contrast to this S-SVM property, the standard SVM solution on the scaled data differs from the orginal solution if scaled back.

tion problem

$$\begin{aligned} \min_{\boldsymbol{w},b} \quad & \frac{1}{2}\,\|\boldsymbol{X}^T\ \boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{Y}\left(\boldsymbol{X}^T\ \boldsymbol{w}\ + b\boldsymbol{1}\right)\ -\ \boldsymbol{1} \geq\ \boldsymbol{0}\ \ . \end{aligned} \tag{10}$$

Note, that $\boldsymbol{X}\ \boldsymbol{X}^T$ is positive semidefinite and above formulation is a convex problem.

The dual formulation is derived in Appendix A as

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T\ \boldsymbol{Y}\ \boldsymbol{X}^*\boldsymbol{X}\ \boldsymbol{Y}\ \boldsymbol{\alpha}\ -\ \boldsymbol{1}^T\ \boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{1}^T\ \boldsymbol{Y}\boldsymbol{\alpha}\ =\ 0\ ,\ \boldsymbol{0}\ \leq\ \boldsymbol{\alpha}\ , \end{aligned} \tag{11}$$

where $\boldsymbol{A}^*$ denotes the pseudo- (Moore-Penrose)-inverse of $\boldsymbol{A}$.

As with standard support vector machines the value b can be computed for constraints where the according $\alpha_i > 0$:

$$\alpha_i > 0: \quad b\ =\ y_i\ -\ \left(\boldsymbol{x}^i\right)^T \boldsymbol{w}\ =\ y_i\ -\ \left(\boldsymbol{x}^i\right)^T \left(\boldsymbol{X}^T\right)^*\ \boldsymbol{Y}\boldsymbol{\alpha}\ , \tag{12}$$

where we used the definition of $\boldsymbol{w}$ in Appendix A

$$\boldsymbol{w}\ =\ \left(\boldsymbol{X}^T\right)^*\ \boldsymbol{Y}\boldsymbol{\alpha}\ , \tag{13}$$

The selected classifier is

$$f(\boldsymbol{x})\ =\ \boldsymbol{x}^T\boldsymbol{w}\ +\ b\ =\ \boldsymbol{x}^T\left(\boldsymbol{X}^T\right)^*\ \boldsymbol{Y}\boldsymbol{\alpha}\ +\ b\ , \tag{14}$$

As shown in Appendix B, our model selection through the S-SVM method is equivalent to sphering and SVM model selection thereafter, if the covariance matrix $\boldsymbol{X}\ \boldsymbol{X}^T$ has full rank. This answers the question "what is the best preprocessing?". However, to integrate sphering and SVM model selection into one framework has several advantages. First, it is more efficient to do both sphering and model selection in one procedure. Secondly, the approach allows to apply the kernel trick and, therefore, carries over to the nonlinear case. Otherwise, sphering in feature space must rely on kernel PCA (Schölkopf and Smola, 2002). Thirdly, our approach is applicable where sphering does not work because the covariance matrix $\boldsymbol{X}\ \boldsymbol{X}^T$ is not invertible. Fourthly, and most importantly, sphering in high dimensional feature spaces with many data points is too expensive in terms of computational time, whereas our approach allows for a fast solver.

In the follwoing we discuss three cases for $\boldsymbol{X}^*$ and the resulting optimization problem. We consider the singular value decomposition of $\boldsymbol{X} \in \mathbb{R}^{N\times L}$:

$$\boldsymbol{X}\ =\ \boldsymbol{U}^T\ \boldsymbol{D}\ \boldsymbol{V} \quad \text{and} \quad \boldsymbol{X}^*\ =\ \boldsymbol{V}^T\ \boldsymbol{D}^*\ \boldsymbol{U}\ , \tag{15}$$

where $\boldsymbol{U}$ is an $\mathbb{R}^{N\times N}$ orthogonal matrix, $\boldsymbol{V}$ is an $\mathbb{R}^{L\times L}$ matrix, $\boldsymbol{D}$ is an $\mathbb{R}^{N\times L}$ diagonal matrix with positive or zero entries, and $\boldsymbol{D}^*$ is the diagonal matrix where the non-zero entries are inverted. We consider the cases (1) $\operatorname{rk}(\boldsymbol{X})\ =\ N$ (e.g. $N < L$), (2) $\operatorname{rk}(\boldsymbol{X})\ =\ L$ (e.g. $L < N$), and (3) $\operatorname{rk}(\boldsymbol{X})\ <\ \min\{L, N\}$.

Case (1) rk$(\boldsymbol{X})\ =\ N$: Because $\left(\boldsymbol{X}\ \boldsymbol{X}^T\right)^{-1}$ exists we obtain $\boldsymbol{X}^*\ =\ \boldsymbol{X}^T\left(\boldsymbol{X}\ \boldsymbol{X}^T\right)^{-1}$. In the objective we obtain $\boldsymbol{X}^*\boldsymbol{X}\ =\ \boldsymbol{V}^T\boldsymbol{I}_{L|N}\boldsymbol{V}$ and the solution is not unique. Here we used

$$\boldsymbol{I}_{L|N}\ :=\ \begin{pmatrix} \boldsymbol{I}_N & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{pmatrix} \tag{16}$$

the N-dimensional identity expanded by zeros to an $L \times L$ matrix. To obtain an unique solution in $\boldsymbol{\alpha}$ a sparseness condition can be imposed on $\boldsymbol{\alpha}$ or the euclidian length of $\boldsymbol{\alpha}$ may be minimized by adding a positive constant to the main diagonal of $\boldsymbol{X}^*\boldsymbol{X}$.

Case (2) $\mathbf{rk}\,(\boldsymbol{X}) \ = \ L$: Because $\left(\boldsymbol{X}^T\ \boldsymbol{X}\right)^{-1}$ exists the pseudo inverse is $X^* \ = \ \left(\boldsymbol{X}^T\ \boldsymbol{X}\right)^{-1}\boldsymbol{X}^T$ and the dual formulation reduces to

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\,\boldsymbol{\alpha}^T\boldsymbol{\alpha} \ - \ \mathbf{1}^T\ \boldsymbol{\alpha} \\ &\text{s.t.} \quad \mathbf{1}^T\ \boldsymbol{Y}\boldsymbol{\alpha} \ = \ 0\ ,\ \mathbf{0} \ \leq \ \boldsymbol{\alpha}\ . \end{aligned} \tag{17}$$

For equal number of training examples in each class we obtain $\boldsymbol{\alpha} = \mathbf{1}$. This dual reflects the fact that PCA in higher dimensions, i.e. if the dimension is larger then the number of data points, orthonormalizes the data points. If all $\boldsymbol{x}^i$ are orthonormal and we set $\boldsymbol{w} \ = \ \sum_{i=1}^{L} y^i\ \boldsymbol{x}^i$ then we obtain the perfect classifier on the training data

$$\langle \boldsymbol{w}, \boldsymbol{x}^i \rangle \ = \ y^i\ . \tag{18}$$

Case (3) $\mathbf{rk}\,(\boldsymbol{X}) \ = J \ \ < \ \min\{L, N\}$: In the objective we obtain $\boldsymbol{X}^*\boldsymbol{X} \ = \ \boldsymbol{V}^T\boldsymbol{I}_{L|J}\boldsymbol{V}$. As in case (1) the solution is not unique.

Now we show explicitely that our apporach is invariant against linear transformation with full rank. We show that for the case that $N \leq L$ (i.e. $\left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1}$ exists), the S-SVM classification values do not change if all data is linear transformed. This is no restriction for high dimensional spaces because our algorithm – and also other SVM algorithms – works in the subspace spanned by the data points, which has dimension L. Therefore our algorithms is invariant against linear transformaitions in the subspace spanned by the data points.

Assume $\boldsymbol{x}$ is mapped to $A\ \boldsymbol{x}$ by a full rank matrix $\boldsymbol{A}$ and assume that $\mathrm{rk}\,(\boldsymbol{X}) \ = \ N$, then the solution of the S-SVM does not change.

$$\left(\boldsymbol{A}_1\ \boldsymbol{A}_1^T\right)^{-1} \ \text{exists:} \quad \Rightarrow \quad \boldsymbol{A}_2\boldsymbol{A}_1\,(\boldsymbol{A}_2\boldsymbol{A}_1)^* \ = \ \boldsymbol{A}_2\boldsymbol{A}_2^* \tag{19}$$

(see page 36 equation (17) in (Lütkepohl, 1996)). Transposing eq. (19) and setting $\boldsymbol{A} \ = \ \boldsymbol{A}_1^T$ and $\boldsymbol{X} \ = \ \boldsymbol{A}_2^T$ gives:

$$(\boldsymbol{A}\boldsymbol{X})^*\,\boldsymbol{A}\boldsymbol{X} \ = \ \ \boldsymbol{X}^*\boldsymbol{X}\ . \tag{20}$$

Because the dual eqs. (11) contains the data only in the term $\boldsymbol{X}^*\boldsymbol{X}$, the solution $\boldsymbol{\alpha}_A$ of the transformed dual is equal to the solution $\boldsymbol{\alpha}$ of the original dual.

We apply again eq. (19)

$$\begin{aligned} &(\boldsymbol{A}\,\boldsymbol{x})^T\ \boldsymbol{w}_A \ = \ \boldsymbol{x}^T\boldsymbol{A}^T\left((\boldsymbol{A}\boldsymbol{X})^T\right)^*\ \boldsymbol{Y}\boldsymbol{\alpha}_A \ = \\ &\boldsymbol{x}^T\left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1}\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{A}^T\left(\boldsymbol{X}^T\boldsymbol{A}^T\right)^*\ \boldsymbol{Y}\boldsymbol{\alpha}_A \ = \\ &\boldsymbol{x}^T\left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1}\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{A}^T\left(\boldsymbol{X}^T\boldsymbol{A}^T\right)^*\right)\ \boldsymbol{Y}\boldsymbol{\alpha}_A \ = \\ &\boldsymbol{x}^T\left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1}\boldsymbol{X}\boldsymbol{X}^T\left(\boldsymbol{X}^T\right)^*\ \boldsymbol{Y}\boldsymbol{\alpha}_A \ \ = \ \boldsymbol{x}^T\boldsymbol{w}\ , \end{aligned} \tag{21}$$

where we used $rk\,(\boldsymbol{X}) \;=\; N$ and, therefore $\left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1}$ exists. Because the α and the values $\boldsymbol{x}^T\boldsymbol{w}$ do not change under linear transformations and the offset b is computed according to eq. (12), the value b does not change, too.

Finally we obtain

$$f_A(\boldsymbol{A}\;\boldsymbol{x}) \;=\; f(\boldsymbol{x})\;, \tag{22}$$

where f_A is the classifier selected by the data linearly transformed by $\boldsymbol{A}$ and f is the classifier selected by the original data. The classification values do not change, i.e. for classification it does not matter how the data is lineareły preprocessed.

For the case that $rk\,(\boldsymbol{X}) \;\leq\; N$, e.g. if $N > L$, and $\boldsymbol{A}$ is not orthogonal, the classification function may change with the transformation. The reason is that the subspace spanned by the training data points changes. For example a data point $\boldsymbol{z}$ which was orthogonal to all training data points $\boldsymbol{X}^T\boldsymbol{z} \;=\; \boldsymbol{0}$ has a classification value of zero because $\boldsymbol{z}^T\left(\boldsymbol{X}^T\right)^* \;=\; \boldsymbol{0}$. With an appropriate $\boldsymbol{A}$ the transformed vector $\boldsymbol{A}\;\boldsymbol{z}$ is no longer orthogonal to all transformed training points.

2.2 Slack Variables

If the data set is not linearly separable then the constraints for correct classification cannot be fulfilled. Using slack variables $\xi_i \;\geq\; 0$ the optimization formulation can obey the constraints:

$$1 - \xi_i \;\leq\; y^i\;\left(\langle\boldsymbol{w}, \boldsymbol{x}^i\rangle \;+\; b\right)\;. \tag{23}$$

That leads to

$$\begin{aligned} \min_{\boldsymbol{w},b,\boldsymbol{\xi}} \quad & \frac{1}{2}\,\|\boldsymbol{X}^T\;\boldsymbol{w}\|_2^2 \;+\; M\;\boldsymbol{1}^T\boldsymbol{\xi} \\ \text{s.t.} \quad & \boldsymbol{Y}\left(\boldsymbol{X}^T\;\boldsymbol{w}\;+b\boldsymbol{1}\right) \;-\; \boldsymbol{1} \;+\; \boldsymbol{\xi} \;\geq\; \boldsymbol{0} \\ & \boldsymbol{\xi} \;\geq\; \boldsymbol{0}\;. \end{aligned} \tag{24}$$

M penalizes wrong classification. The dual formulation is derived in Appendix A as

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T\;\boldsymbol{Y}\;\boldsymbol{X}^*\boldsymbol{X}\;\boldsymbol{Y}\;\boldsymbol{\alpha} \;-\; \boldsymbol{1}^T\;\boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{1}^T\;\boldsymbol{Y}\boldsymbol{\alpha} \;=\; 0\;,\;\boldsymbol{0} \;\leq\; \boldsymbol{\alpha} \;\leq\; M\;\boldsymbol{1}\;. \end{aligned} \tag{25}$$

Note, that the value b is computed through eq. (12) but the condition $0 \;<\; \alpha_i$ must now be replaced by $0 \;<\; \alpha_i \;<\; M$.

2.3 Kernelizing the S-SVM

If the classification problem is nonlinear, the approach from previous section is not sufficient to obtain a nonlinear class boundary. In the case of SVMs a nonlinear class separation boundary is obtained by nonlinear kernels. We do the same here. Using the "kernel trick" we can replace $\boldsymbol{X}^T\ \boldsymbol{X}$ by the Gram (kernel) matrix $\boldsymbol{K}\ =\ \boldsymbol{X}^T\ \boldsymbol{X}$, where the vectors $\boldsymbol{x}^i$ are now from an unknown feature space. However the dot products can be computed through the original vectors $\boldsymbol{o}^i$ via $\langle \boldsymbol{x}^i, \boldsymbol{x}^j \rangle\ =\ K_{ij}\ =\ k\left(\boldsymbol{o}^i, \boldsymbol{o}^j\right)$, where $\boldsymbol{x}^i\ =\ \phi\left(\boldsymbol{o}^i\right)$ and $\boldsymbol{x}^j\ =\ \phi\left(\boldsymbol{o}^j\right)$. As shown in Appendix A the dual eq. (11) can be expressed as

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \boldsymbol{\alpha}^T\ \boldsymbol{Y}\ \boldsymbol{K}^* \boldsymbol{K}\ \boldsymbol{Y}\ \boldsymbol{\alpha}\ -\ \mathbf{1}^T\ \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{1}^T\ \boldsymbol{Y} \boldsymbol{\alpha}\ =\ 0\ ,\ \mathbf{0}\ \leq\ \boldsymbol{\alpha}\ . \end{aligned} \tag{26}$$

In Appendix A the classifier is rewritten as

$$f(\boldsymbol{o})\ =\ \boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o})\ \boldsymbol{K}^*\ \boldsymbol{Y}\ \boldsymbol{\alpha}\ +\ b\ \ , \tag{27}$$

where $\boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o})$ is the vector with components $\boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o})_i\ =\ k\left(\boldsymbol{o}^i, \boldsymbol{o}\right)$.

Again, the value b is computed for constraints where the according $\alpha_i > 0$:

$$\alpha_i > 0:\quad b\ =\ y_i\ -\ \left(\boldsymbol{x}^i\right)^T \boldsymbol{w}\ =\ y_i\ -\ \boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o}^i)\ \boldsymbol{K}^*\ \boldsymbol{Y} \boldsymbol{\alpha}\ . \tag{28}$$

Only Mercer Kernels. For determining $\boldsymbol{K}^*$ it is neccessary that $\boldsymbol{K}$ is positive semidefinite.

Let us consider indefinite kernels which are dot products in Minkowski spaces. A dot product in a Minkowski space can be expressed through a signature matrix $\boldsymbol{D}_s$ which is a diagonal $N \times N$ matrix with ones and minus ones:

$$\boldsymbol{a} \cdot \boldsymbol{b}\ =\ \boldsymbol{a}^T \boldsymbol{D}_S\ \boldsymbol{b}\ . \tag{29}$$

Using the singular value decomposition of $\boldsymbol{X}$,

$$\boldsymbol{X}\ =\ \boldsymbol{U}^T\ \boldsymbol{D}\ \boldsymbol{V}\ , \tag{30}$$

the kernel matrix is

$$\boldsymbol{K}\ =\ \boldsymbol{X}^T\ \boldsymbol{D}_S \boldsymbol{X}\ =\ \boldsymbol{V}^T\ \boldsymbol{D}^T\ \boldsymbol{U} \boldsymbol{D}_S \boldsymbol{U}^T \boldsymbol{D}\ \boldsymbol{V}\ . \tag{31}$$

However, $\boldsymbol{X}^T\ \boldsymbol{X}$ cannot be deduced from $\boldsymbol{K}$ if the signature has positive and negative entries. Therefore, the objective cannot be computed.

If $\left(\boldsymbol{X}^T\ \boldsymbol{X}\right)^{-1}$ exists we can derive a classifer. Setting the derivative of the Lagrangian with respect to $\boldsymbol{w}$ to zero yields

$$\boldsymbol{D}_S\ \boldsymbol{X}\ \boldsymbol{X}^T\ \boldsymbol{D}_S\ \boldsymbol{w}\ =\ \boldsymbol{D}_S\ \boldsymbol{X}\ \boldsymbol{Y} \boldsymbol{\alpha}\ , \tag{32}$$

where the left hand side stems from $\frac{1}{2} \left\|\boldsymbol{X}^T\ \boldsymbol{D}_s\ \boldsymbol{w}\right\|_2^2$ and the right hand side from $\boldsymbol{\alpha}^T \boldsymbol{Y}\ \boldsymbol{X}^T\ \boldsymbol{D}_s\ \boldsymbol{w}$. If we define

$$\boldsymbol{w}\ :=\ \boldsymbol{D}_S\ \left(\boldsymbol{X}^T\right)^*\ \boldsymbol{Y} \boldsymbol{\alpha} \tag{33}$$

then we obtain

$$\begin{aligned} & \boldsymbol{D}_S\ \boldsymbol{X}\ \boldsymbol{X}^T\ \boldsymbol{D}_S\ \boldsymbol{w}\ =\ \boldsymbol{D}_S\ \boldsymbol{X}\ \boldsymbol{X}^T\ \boldsymbol{D}_S\ \boldsymbol{D}_S\ \left(\boldsymbol{X}^T\right)^*\ \boldsymbol{Y} \boldsymbol{\alpha}\ = \\ & \boldsymbol{D}_S\ \boldsymbol{X}\ \boldsymbol{Y} \boldsymbol{\alpha}\ , \end{aligned} \tag{34}$$

where we used $\boldsymbol{D}_S\ \boldsymbol{D}_S\ =\ I_N$ and $\boldsymbol{X}^T\boldsymbol{X}\ \boldsymbol{X}^*\ =\ X^T$. Therefore the $\boldsymbol{w}$ satisfies above equation eq. (32). The classifier is

$$\begin{aligned} f(\boldsymbol{o})\ &=\ \phi(\boldsymbol{o})^T\ \boldsymbol{D}_S\boldsymbol{w}\ =\ \ \boldsymbol{x}^T\ \boldsymbol{D}_S\boldsymbol{w}\ = \\ \boldsymbol{x}^T\left(\boldsymbol{X}^T\right)^*\ \boldsymbol{Y}\boldsymbol{\alpha}\ &=\ \boldsymbol{x}^T\boldsymbol{X}\ \left(\boldsymbol{X}^T\ \boldsymbol{X}\right)^{-1}\boldsymbol{Y}\boldsymbol{\alpha} \end{aligned} \tag{35}$$

and cannot be expressed through the kernel which contains the signature $\boldsymbol{D}_S$. Also the optimization problem

$$\|\boldsymbol{X}^T\boldsymbol{w}\|^2\ =\ \boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{X}^*\boldsymbol{D}_S\boldsymbol{X}\boldsymbol{X}^T\boldsymbol{D}_S\left(\boldsymbol{X}^T\right)^*\ \boldsymbol{Y}\boldsymbol{\alpha} \tag{36}$$

cannot be expressed through the kernel.

3 Regularization and SMO for the Kernel S-SVM

3.1 Regularization

If the kernel defines a dot product in a high dimensional feature space then we face Case (2) from Section 2.1, where $\mathrm{rk}\left(\boldsymbol{X}\right)\ =\ L$. Eq. (18) shows that the resulting classifier is perfect on the training data:

$$\begin{aligned} \boldsymbol{X}^T\boldsymbol{w}\ &=\ \boldsymbol{X}^T\left(\boldsymbol{X}^T\right)^*\ \boldsymbol{Y}\boldsymbol{\alpha}\ = \\ \boldsymbol{X}^T\boldsymbol{X}\left(\boldsymbol{X}^T\ \boldsymbol{X}\right)^{-1}\boldsymbol{Y}\boldsymbol{\alpha}\ &=\ \boldsymbol{Y}\boldsymbol{\alpha}\ =\ \boldsymbol{y}. \end{aligned} \tag{37}$$

The reason for this perfect solution is that sphering data in a space, where the dimension is equal or higher than the number of data points, results in a trivial problem. Each data point is orthogonal to all other data points after sphering. If we set $\boldsymbol{w} = \sum_i \beta_i y_i \tilde{\boldsymbol{x}}^i$ with $\beta_i > 0$ ($\tilde{\boldsymbol{x}}^i$ is $\boldsymbol{x}^i$ after sphering), then we obtain $\left(\tilde{\boldsymbol{x}}^i\right)^T\boldsymbol{w}\ =\ \beta_i y_i$ and we classify all data points correctly.

However overfitting, i.e. high variance in model selection, is very probable. Overfitting results from the fact that our approach is equivalent to sphering and SVM model selection thereafter as shown in Appendix B. Sphering in high dimensional space has high complexity and the resulting SVM problem is trivial. That means overfitting results from the complexity hidden in sphering which also amplifies data directions which contain only noise.

Therefore regularization should not be focused on the model selection part but on the implicit sphering. The idea of regularizing sphering is that directions of the data which are produced through noise should not be scaled to have variance 1 as is done by sphering (weightening). Sphering is based on the inverse of the covariance matrix $\boldsymbol{C}\ =\ \boldsymbol{X}\ \boldsymbol{X}^T$ where the small eigenvalues indicate directions of low data variance which are assumed to stem from noise. We use the fact – as is already known form Kernel PCA Schölkopf and Smola (2002) – that $\boldsymbol{X}\ \boldsymbol{X}^T$ and $\boldsymbol{K}\ =\ \boldsymbol{X}^T\boldsymbol{X}$ have the same non-zero eigenvalues.

One approach to regularization would be to set the p smallest eigenvalues of $\boldsymbol{K}$ to zero. In order to be independent against scaling of the whole space the largest eigenvalue should be normalized to 1. That means the condition (largest eigenvalue divided through the smallest eigenvalue) of the covariance matrix cond($\boldsymbol{C}$) should be small. However this approach has disadvantages. First the regularization hyperparameter is discrete and therefore more sensibel than continuous parameters if the optimal values are estimated, e.g. by cross-validation. Onother, more serious, disadvantage is that a fast solver large kernel matrices is not available.

We regularize the eigenvalues by

$$\nu_{\text{new}} \ = \ (\mathbf{1} - \lambda) \ \nu_{\text{old}} \ + \ \lambda \ , \tag{38}$$

where for $\lambda \ = \ 0$ we obtain the orginal eigenvalue and for $\lambda \ = \ 1$ we obtain as eigenvalue 1. Because of $\frac{\partial \nu_{\text{new}}}{\partial \lambda} \ = \ 1 \ - \ \nu_{\text{old}}$, eigenvalues larger than 1 decrease towards 1 and eigenvalues smaller increase towards 1 if λ is increased from 0 to 1. The condition of the new covariance matrix C is

$$\text{cond}(\boldsymbol{C})_{\text{new}} \ = \ \frac{\nu_{\text{new}}^{\max}}{\nu_{\text{new}}^{\min}} \ = \ \frac{(\mathbf{1} - \lambda) \ \nu_{\text{old}}^{\max} \ + \ \lambda}{(\mathbf{1} - \lambda) \ \nu_{\text{old}}^{\min} \ + \ \lambda} \ = \tag{39}$$
$$\frac{\text{cond}(\boldsymbol{C})_{\text{old}} \ + \ \frac{\lambda}{(\mathbf{1} - \lambda) \ \nu_{\text{old}}^{\min}}}{1 \ + \ \frac{\lambda}{(\mathbf{1} - \lambda) \ \nu_{\text{old}}^{\min}}} \ .$$

For small $\nu_{\text{old}}^{\min}$ we can approximate cond($\boldsymbol{C}$)$_{\text{new}}$ by

$$\text{cond}(\boldsymbol{C})_{\text{new}} \ \approx \ 1 \ + \ \frac{1 - \lambda}{\lambda} \ \nu_{\text{old}}^{\min} \ \text{cond}(\boldsymbol{C})_{\text{old}} \ . \tag{40}$$

It can be seen that the condition number can be much smaller and overfitting less problable.

The eigenvalues of a positive semidefinite matrix can be increased by λ if $\lambda \boldsymbol{I}$ is added to it. That can be seen because $\boldsymbol{I} \ = \ \boldsymbol{U}^T \ \boldsymbol{I} \ \boldsymbol{U}$ holds for an orthogonal matrix $\boldsymbol{U}$ which is obtained from the eigenvalue decomposition of the positiv semidefinite matrix. To lift all eigenvalues of $\boldsymbol{X} \ \boldsymbol{X}^T$ by λ we choose as new objective function

$$\boldsymbol{w}^T \left((1 \ - \ \lambda) \ \boldsymbol{X} \ \boldsymbol{X}^T \ + \ \lambda \ \boldsymbol{I}_N \right) \boldsymbol{w} \ , \tag{41}$$

where I_N is the identity in $\mathbb{R}^{N \times N}$. Note that this regularization is similar to (Schölkopf et al., 1998) and (Chapelle and Schölkopf, 2002) but the regularization terms and the terms to minimimze are exchanged.

The primal formulation with slack variables is

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1 - \lambda}{2} \ \|\boldsymbol{X}^T \ \boldsymbol{w}\|_2^2 \ + \ \frac{\lambda}{2} \ \|\boldsymbol{w}\|_2^2 \ + \ M \ \mathbf{1}^T \boldsymbol{\xi} \\ \text{s.t.} \quad & \boldsymbol{Y} \left(\boldsymbol{X}^T \ \boldsymbol{w} \ + b \mathbf{1} \right) \ - \ \mathbf{1} \ + \ \boldsymbol{\xi} \ \geq \ \mathbf{0} \ , \\ & \boldsymbol{\xi} \ \geq \ \mathbf{0} \ . \end{aligned} \tag{42}$$

With the kernel matrix $\boldsymbol{K} \ = \ \boldsymbol{X}^T \boldsymbol{X}$, we derive in Appendix A as dual

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \boldsymbol{\alpha}^T \ \boldsymbol{Y} \ \left((1 \ - \ \lambda) \ \boldsymbol{K} \ + \ \lambda \ \boldsymbol{I} \right)^{-1} \ \boldsymbol{K} \ \boldsymbol{Y} \ \boldsymbol{\alpha} \ - \ \mathbf{1}^T \ \boldsymbol{\alpha} \\ \text{s.t.} \quad & \mathbf{1}^T \ \boldsymbol{Y} \boldsymbol{\alpha} \ = \ 0 \ , \ \mathbf{0} \ \leq \ \boldsymbol{\alpha} \ \leq \ M \ \mathbf{1} \ , \end{aligned} \tag{43}$$

where I is the identity in $\mathbb{R}^{L \times L}$.

In Appendix A also the expression

$$\boldsymbol{w} \; = \; \left((1 \; - \; \lambda) \, \boldsymbol{X} \; \boldsymbol{X}^T \; + \; \lambda \; \boldsymbol{I}_N\right)^{-1} \; \boldsymbol{X} \boldsymbol{Y} \boldsymbol{\alpha} \; . \tag{44}$$

for the weight vector $\boldsymbol{w}$ and the classifier

$$f(\boldsymbol{o}) \; = \; \boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o}) \; \left((1 \; - \; \lambda) \, \boldsymbol{K} \; + \; \lambda \boldsymbol{I}\right)^{-1} \; \boldsymbol{Y} \; \boldsymbol{\alpha} \; + \; b \;\; . \tag{45}$$

are derived. Here $\boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o})$ is the vector with components $\boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o})_i \; = \; k\left(\boldsymbol{o}^i, \boldsymbol{o}\right)$.

Again, the value b is computed from non-bound α_i:

$$\begin{aligned} & M \; > \; \alpha_i \; > \; 0 : \;\; b \; = \; y_i \; - \; \left(\boldsymbol{x}^i\right)^T \boldsymbol{w} \; = \\ & y_i \; - \; \boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o}^i) \; \left((1 \; - \; \lambda) \, \boldsymbol{K} \; + \; \lambda \boldsymbol{I}\right)^{-1} \; \boldsymbol{Y} \boldsymbol{\alpha} \; . \end{aligned} \tag{46}$$

3.2 Incremental S-SVM with SMO

For large data sets the matrix inversion $\left((1 \; - \; \lambda) \, \boldsymbol{K} \; + \; \lambda \boldsymbol{I}\right)^{-1}$ is too expensive. The inversion is also needed, if the sequential minimal optimization (SMO, Platt, 1999) technique should be applied. In the objective we observe the matrix $\left((1 \; - \; \lambda) \, \boldsymbol{K} \; + \; \lambda \boldsymbol{I}\right)^{-1} \; \boldsymbol{K}$ which can be computed incrementally. With "incrementally" is meant that the expression for n data points can be computed efficiently if the expression for $(n-1)$ data points is known. The efficent computation results from the fact that adding one point adds one row and one column to the Gram matrix $\boldsymbol{K}$.

Therefore we suggest an incremental version of the S-SVM which starts with a working set of two data points and adds step by step one training point to the working set. For each working set of size n the quadratic part $\left((1 \; - \; \lambda) \, \boldsymbol{K}_n \; + \; \lambda \boldsymbol{I}_n\right)^{-1} \; \boldsymbol{K}_n$ is computed from the corresponding expressions $\left((1 \; - \; \lambda) \, \boldsymbol{K}_{n-1} \; + \; \lambda \boldsymbol{I}_n\right)^{-1} \; \boldsymbol{K}_{n-1}$ of the previous working set of size $(n-1)$. The data point which is added has the largest error on the currently selected classifier, i.e. the actual solution. The actual solution is obtained by performing SMO on the working set. The SMO impoves the actual solution such that a new point can be more reliable chosen. In the ideal case the procedure choses only data points which would have been support vectors ($\alpha > 0$) in the original solution and, therefore, the quadratic part is not not the number of data points squared but the number of support vectors squared.

We define

$$\boldsymbol{Q} \; := \; \lambda \boldsymbol{I} \; + \; (1 \; - \; \lambda) \, \boldsymbol{K} \; . \tag{47}$$

$\boldsymbol{Q}$ is a positive definite matrix with eigenvalues $\geq \lambda$, because from $\boldsymbol{K} \; = \boldsymbol{V}^T \; \boldsymbol{D}^T \; \boldsymbol{D} \; \boldsymbol{V}$ it follows that $\boldsymbol{Q} = \boldsymbol{V}^T \; \left(\lambda \boldsymbol{I} \; + \; (1 \; - \; \lambda) \; \boldsymbol{D}^T \; \boldsymbol{D}\right) \; \boldsymbol{V}$.

Adding one training point $\boldsymbol{o}^n$ to $\boldsymbol{K}_{n-1}$ gives

$$\begin{aligned} \boldsymbol{K}_n \; &= \; \begin{pmatrix} \boldsymbol{K}_{n-1} & , & k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right) \\ k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right)^T & , & k(\boldsymbol{o}^n, \boldsymbol{o}^n) \end{pmatrix} \quad \text{and} \\ \boldsymbol{Q}_n \; &= \; \begin{pmatrix} \boldsymbol{Q}_{n-1} & , & (1 \; - \; \lambda) \; k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right) \\ (1 \; - \; \lambda) \; k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right)^T & , & (1 \; - \; \lambda) \; k(\boldsymbol{o}^n, \boldsymbol{o}^n) \; + \; \lambda \end{pmatrix} \; . \end{aligned} \tag{48}$$

If $\lambda > 0$ and if we define

$$\begin{aligned}&\rho \;:=\; (1 \;-\; \lambda)\;\; k(\boldsymbol{o}^n, \boldsymbol{o}^n) \;+\; \lambda \;- \\ &(1 \;-\; \lambda)^2 \;\; k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right)^T \boldsymbol{Q}_{n-1}^{-1} k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right)\end{aligned} \tag{49}$$

then we obtain from the positive definitness of $\boldsymbol{Q}_n$

$$0 \;<\; \det\left(\boldsymbol{Q}_n\right) \;=\; \det\left(\boldsymbol{Q}_{n-1}\right)\; \rho \;, \tag{50}$$

(see Lütkepohl, 1996 (page 50 equation (6a)). It follows that $\rho > 0$. Therefore the inverse of $\boldsymbol{Q}_n$ can be computed according to (Lütkepohl, 1996) (page 29/30 equation (1)) through

$$\begin{aligned}&\boldsymbol{Q}_n^{-1} \;= \\ &\begin{pmatrix} \boldsymbol{Q}_{n-1}^{-1} \;+\; \frac{1}{\rho}\left(1 \;-\; \lambda\right)^2 \;\boldsymbol{Q}_{n-1}^{-1} k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right) k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right)^T \;\; \boldsymbol{Q}_{n-1}^{-1} & , \\ -\; \frac{1}{\rho}\left(1 \;-\; \lambda\right)\;\; k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right)^T \;\; \boldsymbol{Q}_{n-1}^{-1} & , \\ -\frac{1}{\rho}\left(1 \;-\; \lambda\right)\;\; \boldsymbol{Q}_{n-1}^{-1} k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right) & \\ \frac{1}{\rho} & \end{pmatrix} \;.\end{aligned} \tag{51}$$

We define

$$\begin{aligned}&k\left(\boldsymbol{O}_n, \boldsymbol{o}^i\right)^T \;\;:=\;\; \left(k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^i\right)^T, k\left(\boldsymbol{o}^n, \boldsymbol{o}^i\right)\right) \text{ and} \\ &\boldsymbol{\alpha}_n^T \;\;:=\;\; \left(\boldsymbol{\alpha}_{n-1}^T, \alpha_n\right)\end{aligned} \tag{52}$$

then we can compute

$$\begin{aligned}&k\left(\boldsymbol{O}_n, \boldsymbol{o}^i\right)^T \boldsymbol{Q}_n^{-1} \;\;= \\ &\Big(k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^i\right)^T \boldsymbol{Q}_{n-1}^{-1} \;+ \\ &\frac{(1-\lambda)^2}{\rho} k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^i\right)^T \boldsymbol{Q}_{n-1}^{-1} k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right)\;\; k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right)^T \boldsymbol{Q}_{n-1}^{-1} \;\;- \\ &\frac{1-\lambda}{\rho} k\left(\boldsymbol{o}^n, \boldsymbol{o}^i\right) k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right)^T \boldsymbol{Q}_{n-1}^{-1} \;\;\;, \\ &-\frac{1-\lambda}{\rho}\;\; k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^i\right)^T \boldsymbol{Q}_{n-1}^{-1} k\left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right) \;\;+\;\; \frac{1}{\rho}\;\; k\left(\boldsymbol{o}^n, \boldsymbol{o}^i\right)\Big)\end{aligned} \tag{53}$$

and express the new function values according to eq. (27) by

$$f_n\left(\boldsymbol{o}^i\right) \;\;=\;\; k\left(\boldsymbol{O}_n, \boldsymbol{o}^i\right)^T \boldsymbol{Q}_n^{-1} \boldsymbol{Y}_n \boldsymbol{\alpha}_n \;\;+\;\; b_n \;. \tag{54}$$

The function values $f_n\left(\boldsymbol{o}^i\right)$ can be expressed as

$$f_n\left(\boldsymbol{o}^i\right) \;\;=\;\; \sum_{j=1}^{n} y_j \;\alpha_j \;r_{ij} \;\;+\;\; b_n \;, \tag{55}$$

where $r_{ij} \;\;=\;\; \left[k\left(\boldsymbol{O}_n, \boldsymbol{o}^i\right)^T \boldsymbol{Q}_n^{-1}\right]_j$.

If we set

$$F_i \;=\; y_i \; f_n \left(\boldsymbol{o}^i\right) \;-\; 1 \tag{56}$$

we obtain for the Karush-Kuhn-Tucker conditions

$$\begin{aligned} &\alpha_i \;\left(F_i \;+\; \xi_i\right) \;=\; 0 \quad \text{and} \\ &\mu_i \; \xi_i \;=\; 0 \; . \end{aligned} \tag{57}$$

Now we can apply the sequential minimal optimization technique (SMO, Platt, 1999) to optimize or improve the actual solution for the problem with n data points. The first choice for α will be α_n. After some SMO updates we check the KKT conditions of new data points and choose a data point with largest F_i as next point to add.

The computation of $\boldsymbol{Q}_n^{-1} k \left(\boldsymbol{O}_n, \boldsymbol{o}^i\right)$ (note that $k \left(\boldsymbol{O}_n, \boldsymbol{o}^i\right)^T \boldsymbol{Q}_n^{-1} \;=\; \left(\boldsymbol{Q}_n^{-1} k \left(\boldsymbol{O}_n, \boldsymbol{o}^i\right)\right)^T$) can be done in $O(n)$ time if the old values $\boldsymbol{Q}_{n-1}^{-1} k \left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^i\right)$ are stored. Because for all $1 \leq i \leq L$ the $f_n \left(\boldsymbol{x}^i\right)$ must be updated, adding a new point to n training points has complexity of $O(Ln) \;\approx\; O(L^2)$. Note, that the vector $\boldsymbol{Q}_{n-1}^{-1} k \left(\boldsymbol{O}_{n-1}, \boldsymbol{o}^n\right)$ must be computed only once.

The values $\boldsymbol{Q}_n^{-1} k \left(\boldsymbol{O}_n, \boldsymbol{o}^i\right)$ can be used to efficiently describe the classifier eq. (27). For $\lambda > 0$, we obtain from

$$-\;(1 \;-\; \lambda)\, \boldsymbol{Q}^{-1} \boldsymbol{K} \;+\; \boldsymbol{I} \;=\; \boldsymbol{Q}^{-1} \left(\boldsymbol{Q} \;-\; (1 \;-\; \lambda)\, \boldsymbol{K}\right) \;=\; \lambda \boldsymbol{Q}^{-1} \tag{58}$$

the identity

$$\boldsymbol{Q}^{-1} \boldsymbol{K} \;=\; \frac{1}{1-\lambda} \boldsymbol{I} \;-\; \frac{\lambda}{1-\lambda}\, \boldsymbol{Q}^{-1} \; . \tag{59}$$

4 New Bound through the Objective

The new objective upper bounds a bound on the generalization error which is more general than the known bounds (Vapnik, 1995, 1998). These bounds use as input domain a sphere containing the training data which allows for bounding the output range of a function class with bounded $\|\boldsymbol{w}\|$ and, therefore, for bounding the generalization error (Vapnik, 1998; Schölkopf and Smola, 2002). Improved bounds assume a given output range in order to define the function class and implicitely assume an input domain (Shawe-Taylor and Cristianini, 2000).

Here we relax these assumptions on the input domain or output range by estimating how often for a given function the output range is exceeded, rejecting those points, and taking the rejection error into account for the bound.

We consider the function calss $\mathcal{F}$, the set of linear functions $\{f \;\mid\; f(\boldsymbol{x}) \;=\; \langle \boldsymbol{w}, \boldsymbol{x} \rangle \,, \quad \boldsymbol{x} \;\in\; \mathcal{X}\}$ that map from $\mathcal{X}$ to $\mathbb{R}$. The standard function class $\mathcal{F}_0$ is the set of linear functions that map from $\mathcal{X}$ to $[0, 1]$.

We define the empirical output range of f on $\boldsymbol{X}$ as

$$m(f, \boldsymbol{X}) \;:= \max_i \left|\langle \hat{\boldsymbol{w}}, \boldsymbol{x}^i \rangle\right| \; , \tag{60}$$

where $\hat{\boldsymbol{w}} := \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$. We assume that the data is centered around the origin to obtain an thight interval $[-a, a]$ (see Appendix C for how to performe centering).

Let

$$E\left(\mathcal{N}\left(\epsilon, \mathcal{F}, L\right)\right) \ := \ E\left(\mathcal{N}\left(\epsilon, \mathcal{F}, \boldsymbol{X}\right)\right) \tag{61}$$

be the expected ϵ-covering number of the function class $\mathcal{F}$ of L examples with underlying probability P on $\mathcal{X}$ (Shawe-Taylor et al., 1996).

Theorem 1 (Range Bound) *Consider any distribution P on $\mathcal{X}$ from which $\boldsymbol{X} = \left(\boldsymbol{x}^1, \ldots, \boldsymbol{x}^L\right)$ are generated i.i.d. Then with probability $1 - \delta$ over such a L-sample, for any linear classifier f which classifies $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^L$ correctly with margin γ the generalization error is bounded by*

$$\begin{aligned} & \mathcal{B}(L, \frac{m(f, \boldsymbol{X})}{\gamma}, \delta) = \\ & \frac{4}{L}\left(\log_2\left(E\mathcal{N}\left(\frac{\gamma}{2\ m(f, \boldsymbol{X})\ +\ 4\ \gamma}, \mathcal{F}_0, 2L\right)\right)\ + \right. \\ & \left. \log_2\left(\left(\frac{m(f, \boldsymbol{X})}{\gamma}\ +\ 2\right)\frac{8\ L}{\delta}\right)\right)\ . \end{aligned} \tag{62}$$

Proof.

The idea of the proof is, firstly, Step 1, to give a bound for function class with output range a and, secondly, Step 2, compute the probability that a new data point exceeds the output range. The second probability is related to the estimation of the support of the distribution P. Finally, Step 3, both probabilities have to be combined.

Step 1: Bound on the generalization error for bounded output range.

We define another function class $\mathcal{F}_a$ as the set of linear functions from $\mathcal{X}$ to $[-a, a]$. Proposition 19 in (Shawe-Taylor et al., 1996) or Theorem 7.14 in (Shawe-Taylor and Cristianini, 2000) shows that

$$E\left(\mathcal{N}\left(\epsilon, \mathcal{F}_a, L\right)\right) \ = \ E\left(\mathcal{N}\left(\frac{\epsilon}{2a}, \mathcal{F}_0, L\right)\right)\ . \tag{63}$$

Using Theorem 7.7 in (Shawe-Taylor and Cristianini, 2000), Theorem 3.9 in (Shawe-Taylor et al., 1998), and Proposition 19 in (Shawe-Taylor et al., 1996), the generalization error of $f \in \mathcal{F}_{\dashv}$ can be bounded from above with probability $1 - \delta$ by

$$\mathcal{B}_1(L, \frac{a}{\gamma}, \delta) \ = \ \frac{2}{L}\left(\log_2\left(E\mathcal{N}\left(\frac{\gamma}{2\ a}, \mathcal{F}_0, 2L\right)\right)\ +\ \log_2\left(\frac{4\ L\ a}{\delta\ \gamma}\right)\right) \tag{64}$$

provided that the training classification error is zero and all $\boldsymbol{x}$ drawn i.i.d. from the (unknown) distribution $P\mathcal{X}$.

Step 2: Bound on the probability to reject a data point because the output range is exceeded.

The function classes $\mathcal{F}_a$ is sturdy (Shawe-Taylor and Cristianini, 2000), therefore he support bounds in (Schölkopf et al., 1999, 2001) apply. The probability of observing an outlier which is rejected can be bounded:

$$P\{|\langle \boldsymbol{w}, \boldsymbol{x} \rangle| \ \geq \ m \ + \ 2 \ \epsilon\} \ \leq \ \mathcal{B}_2(L, \epsilon, \delta) \ = \tag{65}$$
$$\frac{4}{L} \left(\log_2 E\mathcal{N} \left(\epsilon, \mathcal{F}_a, 2L\right) \ + \ \log_2 \frac{L}{\delta} \right) \ \leq$$
$$\frac{4}{L} \left(\log_2 E\mathcal{N} \left(\frac{\epsilon}{2a}, \mathcal{F}_0, 2L \right) \ + \ \log_2 \frac{L}{\delta} \right) \ .$$

The factor 4 appears because of the absolute value in the bound.

Note, that not all outliers are misclassified, and the trivial random classifier produces 50 % correctly classified examples.

Step 3: Combining both the bound eq. (64) and eq. (65).

We first equalize the the covering numbers of both bounds and then have to compute the confidence that both bounds hold at the same time.

The output range have to be set to $a \ = \ m \ + \ 2 \ \epsilon$ because of the support bound in Step 2. Next, we equalize the ϵ-radius of the balls of the covering numbers from Step 1 and Step 2 and obtain

$$\epsilon \ = \ \gamma \ , \text{thus} \ \ \frac{\gamma}{2 \ a} \ = \ \frac{\gamma}{2 \ m \ + \ 4 \ \gamma} \tag{66}$$

and

$$\frac{a}{\gamma} \ = \ \frac{m \ + \ 2 \ \gamma}{\gamma} \ = \ \frac{m}{\gamma} \ + \ 2 \ . \tag{67}$$

For both steps we get the common expression for the bounds eq. (64) and eq. (65) using

$$\mathcal{B}_{1/2}(L, \frac{a}{\gamma}, \delta) \ = \tag{68}$$
$$\frac{2}{L} \left(\log_2 \left(E\mathcal{N} \left(\frac{\gamma}{2 \ m \ + \ 4 \ \gamma}, \mathcal{F}_0, 2L \right) \right) \ + \ \log_2 \left(\left(\frac{m}{\gamma} \ + \ 2 \right) \frac{4 \ L}{\delta} \right) \right) \ ,$$

where we use as bound in eq. (64) $\mathcal{B}_{1/2}(L, \frac{a}{\gamma}, \delta)$ and in eq. (65) 2 $\mathcal{B}_{1/2}(L, \frac{a}{\gamma}, \delta)$.

We want to compute the confidence that both bounds hold at the same time. However, we must make the assumpution that the confidences in both bounds are independent. The independence assumption is in general wrong because both bounds have the same underlying distribution. However the case that if we have confidence in on bound then we cannot have confidence in the other bound seems not to be very likely.

The probability that both bounds from Step 1 and Step 2 hold simultaneously is

$$(1 - \delta)\,(1 - \delta) = 1 - 2\,\delta + \delta^2 > 1 - 2\,\delta\,. \tag{69}$$

That means we have to replace the δ in the bounds eq. (64) and eq. (65) by $\frac{1}{2}\,\delta$.

The probability of misclassification is now bounded by the sum of two products. The first product is 1 minus the probability of observing an outlier multiplied by the bound on the misclassification. The second product is the trivial misclassification rate of 0.5 multiplied by the bound on observing an outlier:

$$\begin{aligned} &(1 - P\{|\langle \boldsymbol{w}, \boldsymbol{x}\rangle| \geq m\})\ \mathcal{B}_{1/2}\left(L, \frac{a}{\gamma}, \frac{\delta}{2}\right) + 0.5\ 2\ \mathcal{B}_{1/2}\left(L, \frac{a}{\gamma}, \frac{\delta}{2}\right) \\ &\leq 2\ \mathcal{B}_{1/2}\left(L, \frac{a}{\gamma}, \frac{\delta}{2}\right)\ , \end{aligned} \tag{70}$$

where the inequality results from setting $P\{|\langle \boldsymbol{w}, \boldsymbol{x}\rangle| \geq m\} = 0$.

This completes the proof.

■

We obtain our objective if we upper bound our new bound:

$$m/\gamma = \|\boldsymbol{w}\|_2\ \max_i |\langle \hat{\boldsymbol{w}}, \boldsymbol{x}^i\rangle| \leq \sqrt{\sum_i \left(\langle \boldsymbol{w}, \boldsymbol{x}^i\rangle\right)^2} = \|\boldsymbol{X}^T\ \boldsymbol{w}\|_2\ . \tag{71}$$

Our objective is also related to a bound in (Schölkopf and Smola, 2002), where the norm $\|V\|$ of the evaluation operator $V_{\boldsymbol{X}}(\boldsymbol{w}) = \boldsymbol{X}^T\ \boldsymbol{w}$ is used to prove the bound based on entropy numbers. $\|V\|$ can decreased by restricting the possible $\boldsymbol{w}$ to $\|V_{\boldsymbol{X}}(\boldsymbol{w})\|_2^2 \leq R^2$ and, therefore, $\|V\| \leq R$.

5 Experiments

Benchmark experiments on data sets from the UCI benchmark repository which were preprocessed as described in (Rätsch et al., 2001) are documented here. The data sets include "heart" (13 features, 50 training and 100 test points), "flare-solar" (9 features, 100 training and 400 test points), "german" (20 features, 100 training and 300 test points), "image" (18 features, 100 training and 1010 test points), "diabetis" (8 features, 100 training and 300 test points), "splice" (60 features, 100 training and 2175 test points), "thyroid" (5 features, 100 training and 75 test points), and "breast-cancer" (9 features, 100 training and 77 test points). We restricted the training set size to 100 data point in order to obtain larger differences in the results for the compared methods. The data sets were divided into 100 (20 for splice and image) training/test set pairs, where data sets are constructed through resampling where data points were randomly selected for the training set and the remaining data was used for the test set. We downloaded the original 100 training/test set pairs from `http://ida.first.fraunhofer.de/projects/bench/`.

For the S-SVM we fixed the hyperparameter C (the upper bound on the Lagrange multipliers α) to 100 – other values for C as long as they were larger than 2 did not change the results. The other hyperparameters (λ for the S-SVM and ν for the ν-SVM) and the kernel parameter σ (we only tested Gaussian kernels) were chosen with exactly the same procedure for S-SVM and ν-SVM. The same hyperparameter selection procedure was possible because both λ and ν are in $[0, 1]$. The hyperparameters were selected by a 5–fold cross validation on the corresponding training sets. All ν and λ values from $\{0.05, 0.1, 0.15, \ldots, 0.95\}$ were tested. Only 5 σ values were tested. The sigma range was determined by an initial 5-fold cross-validation on the first training set with an ν-SVM and $\nu = 0.5$. First we tested the σ values from $\{0.01, 0.1, 1.0, 10.0, 100.0, 1000.0\}$ and then we scaled the chosen sigma by $\{0.25, 0.5, 1.0, 2.0, 4.0, 8.0\}$ to obtain a first guess. For the 5–fold cross-validation procedure 5 equidistant values in an interval around the first guess were tested for both S-SVM and ν-SVM. The intervals were $[50, 150]$ for heart, $[20, 40]$ for flare-solar, $[20, 40]$ for german, $[20, 40]$ for image, $[10, 30]$ for diabetis, $[60, 100]$ for splice, $[0.3, 3.1]$ for thyroid, and $[40, 60]$ for breast-cancer.

experiment	ν-SVM		S-SVM	
	test error	std test error	test error	std test error
heart	21.50	7.63	**18.79**	4.07
flare-solar	38.54	8.21	**37.95**	5.57
german	**29.01**	3.85	29.14	3.37
image	16.13	4.55	**14.79**	2.98
diabetis	28.39	6.98	**26.03**	2.92
splice	**23.91**	1.99	25.03	2.14
thyroid	**6.29**	2.33	6.99	3.78
breast-cancer	31.27	7.42	**28.36**	4.82

Table 1: Benchmark comparisons between the new S-SVM and the ν-SVM. The columns describe: (1) the data set, (2) average test error for ν-SVM, (3) standard deviation for the ν-SVM test error, (4) average test error for S-SVM, (5) standard deviation for the S-SVM test error. The S-SVM lead in most cases to a lower misclassification error, however the results have low significance as can be seen from the standard deviations. The S-SVM test error standard deviation was on average lower than the test error standard deviation of the ν-SVM.

Table 1 summarizes the results of our experiments on the UCI data sets. The S-SVM yields lower misclassification rate in most cases. However, the results have low significance. An other result is more reliable: the test error standard deviation for the S-SVM is lower. In conclusion, the S-SVM has comparable to slightly better performance on the UCI benchmark dataset but has solutions which do vary less than the solution of the ν-SVM.

For Gaussian kernels the advantage of the S-SVM's scale invariance is not as visible because the feature vectors are implicitly normalized because for Gaussian kernel $k(\boldsymbol{x}, \boldsymbol{x}) = 1$ holds. This equations means that the vectors have length 1. However, we applied the Gaussian kernel to obtain a fair comparison with the ν-SVM.

6 Conclusion

We have introduced a scale invariant framework for the support vector machine technique. We showed how to regularize the new approach and how to speed up the optimization. On experiments the new approach yielded results which are comparable or slightly better than those of the ν-SVM but the S-SVM solutions vary less than those of the ν-SVM. We expect new applications for the S-SVM if new kernels are developed because in this case the vectors may not be normalized in feature space and normalization promised improvement.

A Derivative of the Dual Formulations

In this appendix we will derive the dual formulation and the expressions for $\boldsymbol{w}$ for all the primal optimization problems in the main text.

The most general primal formulation is

$$\begin{aligned} \min_{\boldsymbol{w}, b, \boldsymbol{\xi}} \quad & \frac{1-\lambda}{2} \, \|\boldsymbol{X}^T \, \boldsymbol{w}\|_2^2 \; + \; \frac{\lambda}{2} \, \|\boldsymbol{w}\|_2^2 \; + \; M \, \boldsymbol{1}^T \boldsymbol{\xi} \\ \text{s.t.} \quad & \boldsymbol{Y} \left(\boldsymbol{X}^T \, \boldsymbol{w} \; + b \boldsymbol{1} \right) \; - \; \boldsymbol{1} \; + \; \boldsymbol{\xi} \; \geq \; \boldsymbol{0} \; , \\ & \boldsymbol{\xi} \; \geq \; \boldsymbol{0} \quad . \end{aligned} \tag{72}$$

This primal comprises slack variables, kernels, and regularization. The slack variables $\boldsymbol{\xi}$ can be removed through $M \to \infty$ and pushing them to zero. $\lambda \; = \; 0$ skips the regularization. For the kernel trick we set $\boldsymbol{x}^i \; = \; \phi(\boldsymbol{o}^i)$.

Using the kernel matrix $\boldsymbol{K} \; = \; \boldsymbol{X}^T \boldsymbol{X}$ we obtain as dual

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \boldsymbol{\alpha}^T \, \boldsymbol{Y} \; \left((1 \; - \; \lambda) \, \boldsymbol{K} \; + \; \lambda \, \boldsymbol{I} \right)^* \; \boldsymbol{K} \, \boldsymbol{Y} \, \boldsymbol{\alpha} \; - \; \boldsymbol{1}^T \, \boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{1}^T \, \boldsymbol{Y} \boldsymbol{\alpha} \; = \; 0 \; , \; \boldsymbol{0} \; \leq \; \boldsymbol{\alpha} \; \leq \; M \, \boldsymbol{1} \; , \end{aligned} \tag{73}$$

where I is the identity in $\mathbb{R}^{L \times L}$. For $\lambda > 0$ we have $\left((1 \; - \; \lambda) \, \boldsymbol{K} \; + \; \lambda \, \boldsymbol{I} \right)^* \; = \; \left((1 \; - \; \lambda) \, \boldsymbol{K} \; + \; \lambda \, \boldsymbol{I} \right)^{-1}$. For $\lambda \; = \; 0$ we have $\left((1 \; - \; \lambda) \, \boldsymbol{K} \; + \; \lambda \, \boldsymbol{I} \right)^* \; = \; \boldsymbol{K}^*$ and obtain

$$\boldsymbol{K}^* \boldsymbol{K} \; = \; \left(\boldsymbol{X}^T \boldsymbol{X} \right)^* \boldsymbol{X}^T \boldsymbol{X} \; = \; \boldsymbol{X}^* \left(\boldsymbol{X}^* \right)^T \boldsymbol{X}^T \boldsymbol{X} \; = \; \boldsymbol{X}^* \boldsymbol{X} \; , \tag{74}$$

where we used $\boldsymbol{X}^* \left(\boldsymbol{X}^* \right)^T \boldsymbol{X}^T \; = \; \boldsymbol{X}^*$ (see page 35 equation (9).(f) in (Lütkepohl, 1996)).

The classifier is

$$f(\boldsymbol{o}) \; = \; \boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o}) \; \left((1 \; - \; \lambda) \, \boldsymbol{K} \; + \; \lambda \boldsymbol{I} \right)^* \; \boldsymbol{Y} \, \boldsymbol{\alpha} \; + \; b \quad , \tag{75}$$

where $\boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o})$ is the vector with components $\boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o})_i = k\left(\boldsymbol{o}^i, \boldsymbol{o}\right)$. For $\lambda > 0$ we obtain $((1 - \lambda)\,\boldsymbol{K} + \lambda \boldsymbol{I})^* = ((1 - \lambda)\,\boldsymbol{K} + \lambda \boldsymbol{I})^{-1}$ For $\lambda = 0$ we obtain

$$f(\boldsymbol{o}) = \boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o}) \boldsymbol{K}^* \, \boldsymbol{Y} \, \boldsymbol{\alpha} + b \ , \tag{76}$$

which is in the linear case

$$f(\boldsymbol{x}) = \boldsymbol{x}^T \left(\boldsymbol{X}^T\right)^* \, \boldsymbol{Y} \, \boldsymbol{\alpha} + b \, . \tag{77}$$

The weight vector $\boldsymbol{w}$ can be expressed as

$$\boldsymbol{w} = \left((1 - \lambda)\, \boldsymbol{X} \, \boldsymbol{X}^T + \lambda \, \boldsymbol{I}_N\right)^* \, \boldsymbol{X} \boldsymbol{Y} \boldsymbol{\alpha} \ , \tag{78}$$

where I_N is the identity in $\mathbb{R}^{N \times N}$. For $\lambda > 0$ we obtain $\left((1 - \lambda)\, \boldsymbol{X} \, \boldsymbol{X}^T + \lambda \, \boldsymbol{I}_N\right)^* = \left((1 - \lambda)\, \boldsymbol{X} \, \boldsymbol{X}^T + \lambda \, \boldsymbol{I}_N\right)^{-1}$. For $\lambda = 0$ we obtain

$$\boldsymbol{w} = \left(\boldsymbol{X}^T\right)^* \, \boldsymbol{Y} \boldsymbol{\alpha} \ , \tag{79}$$

where we used $\left(\boldsymbol{X}^*\right)^T \boldsymbol{X}^* \boldsymbol{X} = \left(\boldsymbol{X}^T\right)^*$ (see page 35 equation (9).(f) in (Lütkepohl, 1996)).

The value b is computed from non-bound α_i:

$$\begin{aligned} & M > \alpha_i > 0: \ b = y_i - \left(\boldsymbol{x}^i\right)^T \boldsymbol{w} = \\ & y_i - \boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o}^i) \ ((1 - \lambda)\, \boldsymbol{K} + \lambda \boldsymbol{I})^* \, \boldsymbol{Y} \boldsymbol{\alpha} \, . \end{aligned} \tag{80}$$

Now we want to derive these duals, these classifiers, and these equations for $\boldsymbol{w}$. The Lagrangian of the primal eqs. (72) is

$$\begin{aligned} & L = \frac{1}{2} \boldsymbol{w}^T \left((1 - \lambda) \ \boldsymbol{X} \, \boldsymbol{X}^T + \lambda \boldsymbol{I}_N\right) \, \boldsymbol{w} + M \, \boldsymbol{1}^T \boldsymbol{\xi} - \\ & \boldsymbol{\alpha}^T \left(\boldsymbol{Y} \left(\boldsymbol{X}^T \, \boldsymbol{w} + b \boldsymbol{1}\right) - \boldsymbol{1}\right) - \boldsymbol{\alpha}^T \boldsymbol{\xi} - \boldsymbol{\mu}^T \boldsymbol{\xi} \ , \end{aligned} \tag{81}$$

where $\boldsymbol{\alpha}, \boldsymbol{\mu} \geq \boldsymbol{0}$ are the Lagrange multipliers.

For the optimal solution, the derivatives of the Lagrangian with respect to the primal variables are zero:

$$\begin{aligned} & \nabla_{\boldsymbol{w}} L = \left((1 - \lambda) \ \boldsymbol{X} \, \boldsymbol{X}^T + \lambda\right) \, \boldsymbol{w} - \boldsymbol{X} \boldsymbol{Y} \, \boldsymbol{\alpha} = \boldsymbol{0} \\ & \frac{\partial L}{\partial b} = \boldsymbol{y}^T \boldsymbol{\alpha} = 0 \\ & \nabla_{\boldsymbol{\xi}} L = M \boldsymbol{1} - \boldsymbol{\alpha} - \boldsymbol{\mu} = \boldsymbol{0} \ , \end{aligned} \tag{82}$$

Because $\boldsymbol{\mu} \geq 0$ can be freely chosen, we obtain from the last equation $\alpha_i \leq M$.

The Karush-Kuhn-Tucker (KKT) conditions are:

$$\begin{aligned} &\alpha_i \left(y_i \left(\left(\boldsymbol{x}^i\right)^T \ \boldsymbol{w} \ + b \right) \ - \ 1 \ - \ \xi_i \right) \ = \ 0 \quad \text{and} \\ &\mu_i \ \xi_i \ = \ 0 \ . \end{aligned} \tag{83}$$

If $\alpha_i \ < \ M$ then $\mu_i > 0$ (last equation in eqs. (82)) and, therefore, according to the KKT conditions $\xi_i \ = \ 0$. From $\alpha_i \ > \ 0$ it follows from the KKT conditions $y_i \left(\left(\boldsymbol{x}^i\right)^T \ \boldsymbol{w} \ + b \right) \ - \ 1 \ - \ \xi_i \ = \ 0$. Therefore, we obtain for $0 \ < \ \alpha_i \ < \ M$: $\left(\boldsymbol{x}^i\right)^T \ \boldsymbol{w} \ + b \ - \ y_i \ = \ 0$ which can be solved for b. Thus, the value b is computed from non-bound α_i by

$$\begin{aligned} &M \ > \ \alpha_i \ > \ 0 : \ \ b \ = \ y_i \ - \ \left(\boldsymbol{x}^i\right)^T \boldsymbol{w} \ = \\ &y_i \ - \ \boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o}^i) \ \left((1 \ - \ \lambda) \, \boldsymbol{K} \ + \ \lambda \boldsymbol{I} \right)^* \ \boldsymbol{Y} \boldsymbol{\alpha} \ . \end{aligned} \tag{84}$$

In the follwing we focus on the first equation of eqs. (82) and show that it is fulfilled by

$$\boldsymbol{w} \ = \ \left((1 \ - \ \lambda) \, \boldsymbol{X} \ \boldsymbol{X}^T \ + \ \lambda \ \boldsymbol{I}_N \right)^* \ \boldsymbol{X} \boldsymbol{Y} \boldsymbol{\alpha} \ . \tag{85}$$

A) $\boldsymbol{\lambda \ = \ 0}$**.** We first consider the case $\lambda \ = \ 0$, where the first equation of eqs. (82) reduces to

$$\nabla_{\boldsymbol{w}} L \ = \ \boldsymbol{X} \ \boldsymbol{X}^T \ \boldsymbol{w} \ - \ \boldsymbol{X} \boldsymbol{Y} \ \boldsymbol{\alpha} \ = \ \boldsymbol{0} \ . \tag{86}$$

Using the expression for $\boldsymbol{w}$ in eq. (79) we obtain

$$\begin{aligned} &\boldsymbol{X} \ \boldsymbol{X}^T \ \boldsymbol{w} \ = \ \boldsymbol{X} \ \boldsymbol{X}^T \ \left(\boldsymbol{X}^T\right)^* \ \boldsymbol{Y} \boldsymbol{\alpha} \ = \ \boldsymbol{X} \ \boldsymbol{Y} \boldsymbol{\alpha} \\ &\Longrightarrow \nabla_{\boldsymbol{w}} L \ = \ \boldsymbol{0} \ , \end{aligned} \tag{87}$$

where we used $\boldsymbol{X}^T \boldsymbol{X} \ \boldsymbol{X}^* \ = \ \boldsymbol{X}^T$.

The two equations $\boldsymbol{w} \ = \ \left(\boldsymbol{X}^T\right)^* \ \boldsymbol{Y} \boldsymbol{\alpha}$ and $\boldsymbol{X} \ \boldsymbol{X}^T \ \boldsymbol{w} \ = \ \boldsymbol{X} \ \boldsymbol{Y} \boldsymbol{\alpha}$ allow to replace the primal variables in the Lagrangian. Then we can maximize the Lagrangian with respect to the dual variables and obtain – after multiplication with -1 and, therefore, mimization – the dual formulation

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \quad \frac{1}{2} \boldsymbol{\alpha}^T \ \boldsymbol{Y} \ \boldsymbol{X}^* \boldsymbol{X} \ \boldsymbol{Y} \ \boldsymbol{\alpha} \ - \ \boldsymbol{1}^T \ \boldsymbol{\alpha} \\ &\text{s.t.} \quad \ \ \boldsymbol{1}^T \ \boldsymbol{Y} \boldsymbol{\alpha} \ = \ 0 \ , \ \boldsymbol{0} \ \leq \ \boldsymbol{\alpha} \ \leq \ M \ \boldsymbol{1} \ , \end{aligned} \tag{88}$$

where $\left(A^*\right)^T \ = \ \left(A^T\right)^*$ was used.

For kernels the dual can be expressed through the Gram (kernel) matrix $\boldsymbol{K} \ = \ \boldsymbol{X}^T \ \boldsymbol{X}$, where $K_{ij} \ = \ k\left(\boldsymbol{o}^i, \boldsymbol{o}^j\right)$ and $\boldsymbol{x}^i \ = \ \phi\left(\boldsymbol{o}^i\right)$.

Order to derive an expression based only on kernels we use the singular value decompositions of $\boldsymbol{X}$, $\left(\boldsymbol{X}^T\right)^*$, and $\boldsymbol{K}$:

$$\begin{aligned} &\boldsymbol{X} \ = \ \boldsymbol{U}^T \ \boldsymbol{D} \ \boldsymbol{V} \ , \\ &\left(\boldsymbol{X}^T\right)^* \ = \ \boldsymbol{U}^T \ \left(\boldsymbol{D}^*\right)^T \ \boldsymbol{V} \ , \ \text{and} \\ &\boldsymbol{K} \ = \ \boldsymbol{V}^T \ \boldsymbol{D}^T \ \boldsymbol{D} \ \boldsymbol{V} \ . \end{aligned} \tag{89}$$

Note that $\left(\boldsymbol{D}^T\ \boldsymbol{D}\right)^* \ = \ \boldsymbol{D}^* \left(\boldsymbol{D}^T\right)^*$ and, therefore,

$$\boldsymbol{K}^* \ = \ \boldsymbol{V}^T\ \boldsymbol{D}^* \left(\boldsymbol{D}^T\right)^*\ \boldsymbol{V} \ . \tag{90}$$

We obtain

$$\boldsymbol{X}^* \ = \ \boldsymbol{K}^*\ \boldsymbol{X}^T \ , \tag{91}$$

where we used a property of the pseudo inverse: $\boldsymbol{D}^* \left(\boldsymbol{D}^T\right)^*\ \boldsymbol{D}^T \ = \ \boldsymbol{D}^*$ (see page 35 equation (9).(f) in (Lütkepohl, 1996)). Note, that $\boldsymbol{K}^*$ can be computed through the eigenvalue decomposition of K. Using $\boldsymbol{X}^*\boldsymbol{X} \ = \ \boldsymbol{K}^*\ \boldsymbol{X}^T\boldsymbol{X} \ = \ \boldsymbol{K}^*\boldsymbol{K}$ the dual eq. (88) is

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^T\ \boldsymbol{Y}\ \boldsymbol{K}^*\boldsymbol{K}\ \boldsymbol{Y}\ \boldsymbol{\alpha}\ - \ \boldsymbol{1}^T\ \boldsymbol{\alpha} \\ &\text{s.t.} \quad \boldsymbol{1}^T\ \boldsymbol{Y}\boldsymbol{\alpha} \ = \ 0\ ,\ \boldsymbol{0}\ \leq\ \boldsymbol{\alpha}\ \leq\ M\,\boldsymbol{1}\ , \end{aligned} \tag{92}$$

where $\boldsymbol{K}^*\boldsymbol{K} \ = \ \boldsymbol{V}^T\ \boldsymbol{I}_{L|J}\ \boldsymbol{V}$ (J gives the number of nonzero eigenvalues of $\boldsymbol{K}$).

The classifier is

$$\begin{aligned} &f(\boldsymbol{o}) \ = \ \ \phi(\boldsymbol{o})^T\ \boldsymbol{w}\ + \ b \ = \ \phi(\boldsymbol{o})^T \left(\boldsymbol{X}^T\right)^*\ \boldsymbol{Y}\ \boldsymbol{\alpha}\ + \ b \ = \\ &\phi(\boldsymbol{o})^T\boldsymbol{X}\boldsymbol{K}^*\ \boldsymbol{Y}\ \boldsymbol{\alpha}\ + \ b \ = \ \boldsymbol{k}(\boldsymbol{O},\boldsymbol{o})\boldsymbol{K}^*\ \boldsymbol{Y}\ \boldsymbol{\alpha}\ + \ b \ \ , \end{aligned} \tag{93}$$

where we applied eq. (91) and $\boldsymbol{K}^* \ = \ \left(\boldsymbol{K}^*\right)^T$.

For the linear case we have

$$\begin{aligned} &f(\boldsymbol{x}) \ = \ \boldsymbol{x}^T\boldsymbol{X}\left(\boldsymbol{X}^T\boldsymbol{X}\right)^*\ \boldsymbol{Y}\ \boldsymbol{\alpha}\ + \ b \ = \\ &\boldsymbol{x}^T\boldsymbol{X}\boldsymbol{X}^*\left(\boldsymbol{X}^*\right)^T\ \boldsymbol{Y}\ \boldsymbol{\alpha}\ + \ b \ = \boldsymbol{x}^T\left(\boldsymbol{X}^*\right)^T\ \boldsymbol{Y}\ \boldsymbol{\alpha}\ + \ b \ \ . \end{aligned} \tag{94}$$

B) $\lambda \ > \ 0$. Next we consider the case $\lambda > 0$. For $\lambda \ = \ 1$ we obtain the standard support vector formulation, where above expressions for the dual, $\boldsymbol{w}$, and the classifier are known to be true. Therefore, we assume $\lambda \ < \ 1$.

We need following identity:

$$\begin{aligned} &\left(\lambda\ \boldsymbol{I}_N\ + \ (1\ - \ \lambda)\ \boldsymbol{X}\ \boldsymbol{X}^T\right)^{-1} \ = \\ &\frac{1}{\lambda}\boldsymbol{I}_N\ - \ \frac{1}{\lambda^2}\ \boldsymbol{X}\left(\frac{1}{1\ - \ \lambda}\boldsymbol{I}_L\ + \ \frac{1}{\lambda}\boldsymbol{X}^T\ \boldsymbol{X}\right)^{-1}\ \boldsymbol{X}^T. \end{aligned} \tag{95}$$

where we applied the matrix inversion lemma

$$\begin{aligned} &\left(\boldsymbol{A}_1^{-1}\ + \ \boldsymbol{V}\boldsymbol{A}_2^{-1}\boldsymbol{V}^T\right)^{-1} \ = \\ &\boldsymbol{A}_1\ - \ \boldsymbol{A}_1\ \boldsymbol{V}\ \left(\boldsymbol{A}_2\ + \ \boldsymbol{V}^T\boldsymbol{A}_1\ \boldsymbol{V}\right)^{-1}\ \boldsymbol{V}^T\boldsymbol{A}_1 \end{aligned} \tag{96}$$

with $\boldsymbol{V} \ = \ \boldsymbol{X}$, $\boldsymbol{A}_1 \ = \ \frac{1}{\lambda}\ \boldsymbol{I}_N$, and $\boldsymbol{A}_2 \ = \ \frac{1}{1\ - \ \lambda}\ \boldsymbol{I}_L$.

The first equation of eqs. (82) can be solved for $\boldsymbol{w}$:

$$\boldsymbol{w} \ = \ \left((1\ - \ \lambda)\ \boldsymbol{X}\ \boldsymbol{X}^T\ + \ \lambda\ \boldsymbol{I}\right)^{-1}\ \boldsymbol{X}\boldsymbol{Y}\boldsymbol{\alpha}\ . \tag{97}$$

Using identity eq. (95) the normal vector can be now expressed as

$$\begin{aligned}
&\boldsymbol{w} \;=\; \frac{1}{\lambda}\left(\boldsymbol{I}_N \;-\; \boldsymbol{X}\left(\frac{\lambda}{1\;-\;\lambda}\boldsymbol{I} \;+\; \boldsymbol{X}^T\,\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\right)\boldsymbol{X}\boldsymbol{Y}\boldsymbol{\alpha} \;= \\
&\frac{1}{\lambda}\left(\boldsymbol{X} \;-\; \boldsymbol{X}\left(\frac{\lambda}{1\;-\;\lambda}\boldsymbol{I} \;+\; \boldsymbol{X}^T\,\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{X}\right)\boldsymbol{Y}\boldsymbol{\alpha} \;= \\
&\boldsymbol{X}\frac{1}{\lambda}\left(\boldsymbol{I} \;-\; \left(\frac{\lambda}{1\;-\;\lambda}\boldsymbol{I} \;+\; \boldsymbol{X}^T\,\boldsymbol{X}\right)^{-1}\boldsymbol{X}^T\boldsymbol{X}\right)\boldsymbol{Y}\boldsymbol{\alpha} \;= \\
&\boldsymbol{X}\frac{1}{\lambda}\left(\boldsymbol{I} \;-\; \left(\frac{\lambda}{1\;-\;\lambda}\boldsymbol{I} \;+\; \boldsymbol{K}\right)^{-1}\boldsymbol{K}\right)\boldsymbol{Y}\boldsymbol{\alpha} \;= \\
&\boldsymbol{X}\left(\lambda\boldsymbol{I} \;+\; (1\;-\;\lambda)\,\boldsymbol{K}\right)^{-1}\boldsymbol{Y}\boldsymbol{\alpha}\;,
\end{aligned}\tag{98}$$

where we used in the last equation

$$\begin{aligned}
&(1\;-\;\lambda)\left(\lambda\boldsymbol{I} \;+\; (1\;-\;\lambda)\,\boldsymbol{K}\right)^{-1}\boldsymbol{K} \;= \\
&(1\;-\;\lambda)\left(\lambda\boldsymbol{I} \;+\; (1\;-\;\lambda)\,\boldsymbol{K}\right)^{-1}\left(\boldsymbol{K} \;+\; \frac{\lambda}{1-\lambda}\boldsymbol{I}\right) \;- \\
&(1\;-\;\lambda)\,\frac{\lambda}{1-\lambda}\boldsymbol{I}\left(\lambda\boldsymbol{I} \;+\; (1\;-\;\lambda)\,\boldsymbol{K}\right)^{-1} \;= \\
&\boldsymbol{I} \;-\; \lambda\left(\lambda\boldsymbol{I} \;+\; (1\;-\;\lambda)\,\boldsymbol{K}\right)^{-1}\;.
\end{aligned}\tag{99}$$

The first equation of eqs. (82) is

$$\left((1-\lambda)\;\boldsymbol{X}\;\boldsymbol{X}^T \;+\; \lambda\boldsymbol{I}_N\right)\;\boldsymbol{w} \;=\; \boldsymbol{X}\boldsymbol{Y}\;\boldsymbol{\alpha}\;. \tag{100}$$

Using this equation and the equation eq. (98) we obtain

$$\begin{aligned}
&\boldsymbol{w}^T\left((1-\lambda)\;\boldsymbol{X}\;\boldsymbol{X}^T \;+\; \lambda\boldsymbol{I}_N\right)\;\boldsymbol{w} \;=\; \boldsymbol{\alpha}^T\boldsymbol{Y}\boldsymbol{X}^T\;\boldsymbol{w} \;= \\
&\boldsymbol{\alpha}^T\;\boldsymbol{Y}\;\left((1\;-\;\lambda)\,\boldsymbol{K} \;+\; \lambda\;\boldsymbol{I}\right)^{-1}\;\boldsymbol{K}\;\boldsymbol{Y}\;\boldsymbol{\alpha}\;,
\end{aligned}\tag{101}$$

where we used

$$\left((1\;-\;\lambda)\,\boldsymbol{K} \;+\; \lambda\;\boldsymbol{I}\right)^{-1}\;\boldsymbol{K} \;=\; \boldsymbol{K}\;\left((1\;-\;\lambda)\,\boldsymbol{K} \;+\; \lambda\;\boldsymbol{I}\right)^{-1} \tag{102}$$

which follows from eq. (99) and

$$\begin{aligned}
&\boldsymbol{K}\,(1\;-\;\lambda)\left(\lambda\boldsymbol{I} \;+\; (1\;-\;\lambda)\,\boldsymbol{K}\right)^{-1} \;= \\
&\left(\boldsymbol{K} \;+\; \frac{\lambda}{1-\lambda}\boldsymbol{I}\right)(1\;-\;\lambda)\left(\lambda\boldsymbol{I} \;+\; (1\;-\;\lambda)\,\boldsymbol{K}\right)^{-1} \;- \\
&(1\;-\;\lambda)\,\frac{\lambda}{1-\lambda}\boldsymbol{I}\left(\lambda\boldsymbol{I} \;+\; (1\;-\;\lambda)\,\boldsymbol{K}\right)^{-1} \;= \\
&\boldsymbol{I} \;-\; \lambda\left(\lambda\boldsymbol{I} \;+\; (1\;-\;\lambda)\,\boldsymbol{K}\right)^{-1}\;.
\end{aligned}\tag{103}$$

Analog to the case $\lambda \;=\; 0$ we derive for the dual formulation

$$\begin{aligned}
&\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\boldsymbol{\alpha}^T\;\boldsymbol{Y}\;\left((1\;-\;\lambda)\,\boldsymbol{K} \;+\; \lambda\;\boldsymbol{I}\right)^{-1}\;\boldsymbol{K}\;\boldsymbol{Y}\;\boldsymbol{\alpha} \;-\; \boldsymbol{1}^T\;\boldsymbol{\alpha} \\
&\text{s.t.} \quad\;\; \boldsymbol{1}^T\;\boldsymbol{Y}\boldsymbol{\alpha} \;=\; 0\;,\; \boldsymbol{0} \;\leq\; \boldsymbol{\alpha} \;\leq\; M\;\boldsymbol{1}\;.
\end{aligned}\tag{104}$$

Note, that for $\lambda = 1$ we obtain the standard SVM dual. Because of eq. (102) we know that $((1 - \lambda) \boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{K}$ is symmetric and positive semidefinite.

The classifier can be computed by applying eq. (98):

$$\begin{aligned} & f(\boldsymbol{o}) = \phi(\boldsymbol{o})^T \boldsymbol{w} + b = \boldsymbol{x}^T \boldsymbol{w} + b = \\ & \boldsymbol{x}^T \boldsymbol{X} ((1 - \lambda) \boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{Y} \boldsymbol{\alpha} + b = \\ & \boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o}) ((1 - \lambda) \boldsymbol{K} + \lambda \boldsymbol{I})^{-1} \boldsymbol{Y} \boldsymbol{\alpha} + b . \end{aligned} \tag{105}$$

Again, we observe for $\lambda = 1$ the standard SVM classifier:

$$f(\boldsymbol{o}) = \boldsymbol{k}(\boldsymbol{O}, \boldsymbol{o}) \boldsymbol{Y} \boldsymbol{\alpha} . \tag{106}$$

B Sphering

We show that for the case that $N \leq L$ (i.e. $\left(\boldsymbol{X}\boldsymbol{X}^T\right)^{-1}$ exists), the S-SVM classification is equal to sphering and applying the SVM thereafter. In (Schölkopf et al., 1998) a similar property of the covariance matrix of tangent vectors was shown.

For sphered data, i.e. $\boldsymbol{X} \boldsymbol{X}^T = I$, we have $\|\boldsymbol{X}^T \boldsymbol{w}\|_2^2 = \boldsymbol{w}^T \boldsymbol{X} \boldsymbol{X}^T \boldsymbol{w} = \boldsymbol{w}^T \boldsymbol{w} = \|\boldsymbol{w}\|_2^2$. That means our objective is the classical SVM objective.

In the following we analyze the difference between our S-SVM and standard SVM on sphered data. The covariance matrix $\boldsymbol{C}$ is approximated by the data matrix $\boldsymbol{X}$ via $\boldsymbol{C} \approx \boldsymbol{X} \boldsymbol{X}^T$. We assume that $\boldsymbol{C}^{-1} = \left(\boldsymbol{X} \boldsymbol{X}^T\right)^{-1}$ exists, thus $X^* = \boldsymbol{X}^T \left(\boldsymbol{X} \boldsymbol{X}^T\right)^{-1}$. We rescale the data in such a way that it has spherical shape: $\tilde{\boldsymbol{X}} = \boldsymbol{C}^{-\frac{1}{2}} \boldsymbol{X}$, where $\tilde{\boldsymbol{X}}$ is the spherical data. The classical support vector formulation applied to $\tilde{\boldsymbol{X}}$ is

$$\begin{aligned} \min_{\boldsymbol{w}, b} \quad & \frac{1}{2} \|\boldsymbol{w}\|_2^2 \\ \text{s.t.} \quad & \boldsymbol{Y} \left(\boldsymbol{X}^T \left(\boldsymbol{X} \boldsymbol{X}^T\right)^{-\frac{1}{2}} \boldsymbol{w} + b\boldsymbol{1} \right) - \boldsymbol{1} \geq \boldsymbol{0} , \end{aligned} \tag{107}$$

where we used $\left(\boldsymbol{X} \boldsymbol{X}^T\right)^{-T} = \left(\boldsymbol{X} \boldsymbol{X}^T\right)^{-1}$.

We look at the derivative of the Lagrangian L with respect to $\boldsymbol{w}$:

$$\nabla_{\boldsymbol{w}} L = \boldsymbol{w} - \left(\boldsymbol{X} \boldsymbol{X}^T\right)^{-\frac{1}{2}} \boldsymbol{X} \boldsymbol{Y} \boldsymbol{\alpha} = \boldsymbol{0} . \tag{108}$$

We obtain

$$\|\boldsymbol{w}\|_2^2 = \boldsymbol{w}^T \boldsymbol{w} = \boldsymbol{\alpha}^T \boldsymbol{Y} \boldsymbol{X}^T \left(\boldsymbol{X} \boldsymbol{X}^T\right)^{-1} \boldsymbol{X} \boldsymbol{Y} \boldsymbol{\alpha} . \tag{109}$$

and the dual formulation

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{Y} \boldsymbol{X}^* \boldsymbol{X} \boldsymbol{Y} \boldsymbol{\alpha} - \boldsymbol{1}^T \boldsymbol{\alpha} \\ \text{s.t.} \quad & \boldsymbol{1}^T \boldsymbol{Y} \boldsymbol{\alpha} = 0 , \ \boldsymbol{\alpha} \geq \boldsymbol{0} \end{aligned} \tag{110}$$

That is exactly the optimization formulation for the sphered SVM eqs. (11).

Comparing eq. (108) with eq. (13) we see that only how $\boldsymbol{w}$ is computed distinguishes the S-SVM and sphering. We obtain

$$\boldsymbol{w}_{\mathrm{S-SVM}} \;=\; \left(\boldsymbol{X}\,\boldsymbol{X}^T\right)^{-\frac{1}{2}} \boldsymbol{w}_{\mathrm{sphering}} \;=\; \boldsymbol{C}^{-\frac{1}{2}} \boldsymbol{w}_{\mathrm{sphering}} \; . \qquad (111)$$

In Subsection 9.2.4 of (Smola, 1998) an approach (convex combination algorithm) is presented where the normal vector points in the directions where the the data is spread most. However, the S-SVM weight vector points in directions where the data is spread least as can be seen in above equation.

The nonzero eigenvalues of the kernel matrix $\boldsymbol{X}^T\,\boldsymbol{X}$ and the nonzero eigenvalues of the covariance matrix $\boldsymbol{X}\,\boldsymbol{X}^T$ are equal which can easily seen through the singular value decomposition. For ”kernel principal component analysis” (Schölkopf and Smola, 2002) the same property is utilized.

C Centering at the Origin

In Appendix 4 the data was assumed to be centered around the origin $\mathbf{0}$. In this section this assumption is justified by showning how to center the data and, therefore, how to obtain a translation invariant dual formulation. The centering is performed by multiplying the data $\boldsymbol{X}$ by the matrix $\boldsymbol{B}$:

$$\boldsymbol{B} \;:=\; \boldsymbol{I} \;-\; \frac{1}{L}\,\mathbf{1}\,\mathbf{1}^T \qquad (112)$$
$$\boldsymbol{B}\,\boldsymbol{B} \;=\; \boldsymbol{B}\ (\text{idempotent})\ ,\quad \boldsymbol{B}^T \;=\; \boldsymbol{B}\ ,\quad \boldsymbol{B}^* \;=\; \boldsymbol{B}\ .$$

We apply our method to the centered data $\boldsymbol{X}\,\boldsymbol{B}$.

Our method is now translation invariant. That can be seen for a translation $\boldsymbol{z}$ of the data $\boldsymbol{X}$ which results in $\boldsymbol{X} \;+\; \boldsymbol{z}\,\mathbf{1}^T$ and

$$\left(\boldsymbol{X} \;+\; \boldsymbol{z}\,\mathbf{1}^T\right)\,\boldsymbol{B} \;= \qquad (113)$$
$$\boldsymbol{X}\,\boldsymbol{B} \;+\; \left(\boldsymbol{z}\,\mathbf{1}^T \;-\; \frac{1}{L}\,\boldsymbol{z}\,\left(1^T\,\mathbf{1}\right)\mathbf{1}^T\right) \;=\; \boldsymbol{X}\,\boldsymbol{B}\ .$$

For the case that $\left(\boldsymbol{X}^T\,\boldsymbol{X}\right)^{-1}$ exists, the dual does not change. That can be shown by

$$\left(\boldsymbol{X}\,\boldsymbol{B}\right)^*\,\left(\boldsymbol{X}\,\boldsymbol{B}\right) \;=\; \left(\boldsymbol{B}\,\boldsymbol{X}^T\right)\,\left(\boldsymbol{B}\,\boldsymbol{X}^T\right)^* \;=\; \boldsymbol{B}\,\boldsymbol{B}^* \;=\; \boldsymbol{B}\ . \qquad (114)$$

For the second ”=” we used the fact that $\left(\boldsymbol{X}^T\,\boldsymbol{X}\right)^{-1}$ exists and eq. (19). The condition that

$$\mathbf{1}^T\,\boldsymbol{Y}\,\boldsymbol{\alpha} \;=\; \mathbf{0} \qquad (115)$$

proves that

$$\boldsymbol{B}\,\boldsymbol{Y}\,\boldsymbol{\alpha} \;=\; \left(\boldsymbol{I} \;-\; \frac{1}{L}\,\mathbf{1}\,\mathbf{1}^T\right)\,\boldsymbol{Y}\,\boldsymbol{\alpha} \;=\; \boldsymbol{Y}\,\boldsymbol{\alpha}\ . \qquad (116)$$

Now we obtain for the quadratic part $\frac{1}{2}\boldsymbol{\alpha}^T\,\boldsymbol{Y}\,\left(\boldsymbol{X}\boldsymbol{B}\right)^*\left(\boldsymbol{X}\boldsymbol{B}\right)\,\boldsymbol{Y}\,\boldsymbol{\alpha}$ of the dual the expression $\boldsymbol{\alpha}^T\boldsymbol{Y}\,\boldsymbol{B}\,\boldsymbol{Y}\,\boldsymbol{\alpha} \;=\; \boldsymbol{\alpha}^T\,\boldsymbol{\alpha}$. Therefore, the solution of the dual formulation (17) does not change if the data is centered. For kernels centering can be performed via $\boldsymbol{B}\boldsymbol{K}\boldsymbol{B}$ as for kernel PCA (Schölkopf and Smola, 2002).

References

C. M. Bishop. *Neural Networks for Pattern Recognition.* Clarendon Press, Oxford, 1995.

B. E. Boser, I. M. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In D. Haussler, editor, *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152, Pittsburgh, PA, 1992. ACM Press.

O. Chapelle and B. Schölkopf. Incorporating invariances in nonlinear SVMs. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2002.

R. Herbrich and T. Graepel. A PAC-bayesian margin bound for linear classifiers: Why SVMs work. In *Advances in Neural Information Processing Systems*, volume 12, pages 224–230, 2000.

H. Lütkepohl. *Handbook of Matrices.* John Wiley & Sons, 1996.

J. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning*, pages 185–208, Cambridge, MA, 1999. MIT Press.

G. Rätsch, T. Onoda, and K.-R. Müller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001. Also: NeuroCOLT Technical Report 1998-021.

B. Schölkopf, J. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. *Neural Computation*, 13:1443–1471, 2001.

B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Estimating the support of a high-dimensional distribution. Technical Report 99-87, Microsoft Research, 1999.

B. Schölkopf, P. Simard, A. J. Smola, and V. Vapnik. Prior knowledge in support vector kernels. In M. Jordan, M. Kearns, and S. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 640–646, Cambridge, MA, 1998. MIT Press.

B. Schölkopf and A. J. Smola. *Learning with kernels – Support Vector Machines, Reglarization, Optimization, and Beyond.* MIT Press, Cambridge, 2002.

J. Shawe-Taylor, P. L. Bartlett, R. Williamson, and M. Anthony. A framework for structural risk minimization. In *Proceedings of the 9th Annual Conference on Computational Learning Theory*, pages 68–76, New York, 1996. Association for Computing Machinery.

J. Shawe-Taylor, P. L. Bartlett, R. C. Williamson, and M. Anhtony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory*, 44(5):1926–1940, 1998.

J. Shawe-Taylor and N. Cristianini. On the generalisation of soft margin algorithms. Technical Report NC2-TR-2000-082, NeuroCOLT2, Department of Computer Science, Royal Holloway, University of London, 2000.

A. J. Smola. *Learning with kernels.* PhD thesis, Technische Univ. Berlin, 1998.

V. N. Vapnik. *The Nature of Statistical Learning Theory.* Springer Verlag, New York, 1995. ISBN 0-387-94559-8.

V. N. Vapnik. *Statistical Learning Theory.* Adaptive and learning systems for signal processing, communications, and control. Wiley, New York, 1998.