Classification, Regression, and Feature Selection on Matrix Data

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

Technischer Bericht 2004/02
ISSN 1436-9915

Contents

Classification, Regression, and Feature Selection on Matrix Data

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

Abstract

We describe a new technique for the analysis of data which is given in matrix form. We consider two sets of objects, the "row" and the "column" objects, and we represent these objects by a matrix of numerical values which describe their mutual relationships. We then introduce a new technique, the "Potential Support Vector Machine" (P-SVM), as a large-margin based method for the construction of classifiers and regression functions for the "column" objects. Contrary to standard support vector machine (SVM) approaches, the P-SVM minimizes a scale-invariant capacity measure under a new set of constraints. As a result, the P-SVM can handle data matrices which are neither positive definite nor square, and leads to a usually sparse expansion of the classification boundary or the regression function in terms of the "row" rather than the "column" objects. We introduce two complementary regularization schemes in order to avoid overfitting for noisy data sets. The first scheme improves generalization performance for classification and regression problems, the second scheme leads to the selection of a small and informative set of "row" objects and can be applied to feature selection. A fast optimization algorithm based on the "Sequential Minimal Optimization" (SMO) technique is provided.

We first apply the new method to so-called pairwise data, i.e. "row" and "column" objects are from the same set. Pairwise data can be represented in two ways. The first representation uses vectorial data and constructs a Gram matrix from feature vectors using a kernel function. Benchmark results show, that the P-SVM method provides superior classification and regression results and has the additional advantages that kernel functions are no longer restricted to be positive definite. The second representation uses a measured matrix of mutual relations between objects rather than vectorial data. The new classification and regression method performs very well compared to standard techniques on benchmark data sets. More importantly, however, experiments show that the P-SVM can be very effectively used for feature selection. Then we apply the P-SVM to genuine matrix data, where "row" and "column" objects

are from different sets, and, again, the data matrix is either constructed via a kernel function combining "row" and "column" objects or obtained by measurements. On various benchmark data sets we demonstrate the new method's excellent performance for classification, regression, and feature selection problems. For both pairwise and matrix data benchmarks are performed not only with toy data, but also with several real world data sets including data from the UCI repository, protein classification, web-page classification, and DNA microarray data.

1 Introduction

Learning from examples in order to predict is one of the standard tasks in machine learning. Many techniques have been developed to solve what statisticians call classification and regression problems, but by far most of them were specifically designed for vectorial data. Vectorial data, where data objects are described by vectors of numbers and where these data vectors are treated as elements of a vector space, are very convenient, because of the structure imposed by the typically chosen Euclidean metric. However, for many datasets a vector-based description is inconvenient or simply wrong, and other representations like matrices, trees, or graphs, which take relationships between objects into account, are often more appropriate.

a) Pairwise Data $\boldsymbol{K}^\top$

	A	B	C	D	E	F	G	H	I	J	K	L
A	0.9	−0.1	−0.8	0.5	0.2	−0.5	−0.7	−0.9	0.2	−0.7	0.4	−0.3
B	−0.1	0.9	0.6	0.3	−0.7	−0.6	0.3	0.7	−0.3	−0.8	−0.7	−0.9
C	−0.8	0.6	0.9	0.2	−0.6	0.6	0.5	0.2	−0.7	−0.5	−0.1	0.6
D	0.5	0.3	0.2	0.9	0.7	0.1	0.3	−0.1	0.6	0.9	−0.9	−0.1
E	0.2	−0.7	−0.6	0.7	0.9	−0.9	−0.5	0.4	0.1	−0.3	−0.6	0.7
F	−0.5	−0.6	0.6	0.1	−0.9	0.9	0.9	−0.2	−0.6	−0.5	−0.4	−0.3
G	−0.7	0.3	0.5	0.3	−0.5	0.9	0.9	−0.3	−0.3	0.6	0.9	−0.7
H	−0.9	0.7	0.2	−0.1	0.4	−0.2	−0.3	0.9	0.2	−0.9	0.3	0.4
I	0.2	−0.3	−0.7	0.6	0.1	−0.6	−0.3	0.2	0.9	−0.3	−0.7	0.8
J	−0.7	−0.8	−0.5	0.9	−0.3	−0.5	0.6	−0.9	−0.3	0.9	−0.1	−0.5
K	0.4	−0.7	−0.1	−0.9	−0.6	−0.4	0.9	0.3	−0.7	−0.1	0.9	0.1
L	−0.3	−0.9	0.6	−0.1	0.7	−0.3	−0.7	0.4	0.8	−0.5	0.1	0.9

b) Matrix Data $\boldsymbol{K}^\top$

	A	B	C	D	E	F	G
α	1.3	−2.2	−1.6	7.8	6.6	−7.5	−4.8
β	−1.8	−1.1	7.2	2.3	9.0	3.8	3.9
χ	1.2	1.9	−2.9	−2.2	−4.4	−4.7	−8.4
δ	3.7	0.8	−0.6	2.5	−5.7	0.1	−0.3
ε	9.2	−9.4	−8.3	9.2	−2.4	−3.9	1.9
ϕ	−7.7	8.6	−9.7	−7.4	2.6	6.9	2.9
γ	−4.8	0.1	−1.2	0.9	0.2	2.7	0.2
η	0.7	−1.7	0.3	−7.2	−1.8	4.6	2.6
ι	−6.2	−6.2	1.8	3.6	−0.7	−9.4	0.9
φ	9.0	4.8	−8.3	−0.8	−2.0	4.4	−1.9
κ	6.2	9.0	1.5	−1.1	7.7	8.4	−2.1
λ	9.6	7.0	2.5	−4.3	−5.4	0.7	1.2

Figure 1: Pairwise data (a) and matrix data (b). For explanation see text.

In the following we will study representations of data objects which are based on matrices. The description consists of two sets of objects: "column" objects and "row" objects (Fig. 1b). "Column" objects are the objects to be

described, while "row" objects are the objects which serve for their description. Every row-column pair is described by a number. The whole dataset can thus be represented using a rectangular matrix, like in an Excel sheet, whose entries denote the relationships between the "row" and the "column" objects. In the following we will call representations of this form *matrix data*. A special case occurs if "row" and "column" objects are from the same set (Fig. 1a). In this case we will call the representation *pairwise data*, and the entries of the matrix can often be interpreted as the degree of similarity (or dissimilarity) between pairs of objects.

Matrix-based descriptions are more powerful and more flexible than vector-based descriptions, but vectorial data can always be brought into matrix form, when required. This is usually done in the context of kernel-based classifiers or regression functions (Schölkopf and Smola, 2002; Vapnik, 1998), for example when using the support vector machine technique. Before the predictor is learned from examples, a Gram matrix of mutual similarities is calculated by applying a kernel function to pairs of feature vectors. The result is a pairwise data matrix (cf. Fig. 1a), which is then used for learning the predictor. A similar procedure can also be used in the case where the "row" and "column" objects are from different sets (cf. Fig. 1b). If both of them are described by feature vectors, a matrix can be calculated by applying a kernel function to pairs of feature vectors, one vector describing a "row" and the other vector describing a "column" object. One example for this is the drug-gene matrix of Scherf et al. (2000), which was constructed as the product of a measured drug-sample and a measured sample-gene matrix and where the kernel function was a scalar product. However, matrix-based descriptions are most useful if the matrix entries are measured directly.

Pairwise data representation can be found in many datasets which are generated by determining how similar objects from *one set* are. Examples from the bioinformatics domain include similarities of protein sequences (Lipman and Pearson, 1985), biophysically defined similarities between proteins (Sigrist et al., 2002; Falquet et al., 2002), gene similarity measure based on their chromosome location (Cremer et al., 1993; Lu et al., 1994), or co-expression data for genes (Heyer et al., 1999). Other application areas are document processing and web-mining, where pairwise data arise for example in the form of co-citation matrices for text documents (White and McCain; Bayer et al.; Ahlgren et al.), or binary connectivity matrices which summarize the presence and absence of hyperlinks between web-pages (Kleinberg, 1999). In general, however, measured matrices are symmetric but may no longer be positive definite, and even if they are for the training set, they may not remain positive definite if new examples are included.

Genuine *matrix data* — in contrast to previous examples for pairwise data — are observed in many datasets, where *two sets* of objects are related by pairwise measurements. One prominent example from the bioinformatics domain are DNA microarray data (Southern, 1988; Lysov et al., 1988; Drmanac et al., 1989; Bains and Smith, 1988). Here the "column" objects are tissue or cell-line samples which are described by a set of "row" objects, the genes. For every

sample-gene pair a number is measured, which is related to the expression level of this particular gene in this particular sample. These values are then summarized in a matrix. Another example are web-documents, where the "column" objects are web-pages which are described by whether other web-pages, the "row" objects, contain a hyperlink reference. Every pair of column and row web page is then characterized by the number of directed hyperlinks from row to column, which gives rise to a rectangular matrix of ordinal values[1]. There are many further examples for matrix data. Images ("column" objects) can be described by the scalar values (matrix elements) obtained from average linear or non-linear filter responses ("row" objects) to an image. Similarly, time-series can be described by scalar values which may be the components of their short term power spectra, wavelet coefficients, or components of the autocorrelation functions. Customers of a company can be described by their product preferences or by their transaction data, documents in a database can be described by word-frequencies, and molecules can be described by transferable atom equivalent (TAE) descriptors (Mazza et al., 2001), for the purpose of drug design. Traditionally, "row" objects have been called "features" and "column" vectors of the data matrix have mostly been treated as "feature vectors" which live in a vector space. Difficulties, however, arise when the features are heterogeneous, and apples and oranges must be compared. Even more difficulties arise for the special case of pairwise data discussed before, for which descriptive and described objects are actually the same and for which it is hard to justify any differentiation between objects and features.

Classification and regression problems on matrix data, i.e. the task to learn predictors for attributes of the "column" objects, have been mostly addressed within the feature vector framework — despite abovementioned problems, e.g. despite the fact that the distinction between features and objects is blurred for pairwise data (Graepel et al., 1999; Mangasarian, 1998). An approach to pairwise data which does not use vectorial representations is to interpret the pairwise relation data matrix as a Gram matrix and to apply support vector machines (SVM) for classification and regression if the data matrix is positive semidefinite (Graepel et al., 1999). For indefinite (but symmetric) matrices two non-vectorial approaches have been suggested (Graepel et al., 1999). In the first approach, the data matrix is made positive definite by projecting into the subspace spanned by the eigenvectors with positive eigenvalues. Clearly, this is an approximation and one would expect it to give good results only, if the absolute values of the negative eigenvalues are small compared to the dominant positive ones. The other approach is also based on an eigenvalue decomposition and treats directions of negative eigenvalues by just flipping the sign of these eigenvalues. This approach, however, lacks a theoretical foundation. All these non-vectorial approaches are restricted to pairwise data and lead to a matrix of object relations which is positive semidefinite, but they do not ensure that positive semidefiniteness still holds, if a new test object must be included.

[1] Note, that in previous paragraph for pairwise data examples the linking matrix was symmetric because links were considered bidirectional. Here the links are undirectional and the data is no longer pairwise because it is not symmetric.

Therefore they may fail in the test phase. Another embedding approach was suggested by Herbrich et al. (1998) for antisymmetric matrices, but this was specifically designed for data sets, where the matrix entries denote preference relations between objects, for example with respect to the difference in relevance of two documents to a particular user of a document database. So far, no general method exists for learning classifiers or regression functions from data represented in matrix form.

In this contribution we argue that — in order to avoid abovementioned shortcomings — it is beneficial to consider "column" and "row" objects on equal footing. With this we mean, that the construction of the data matrix or the actual measurement of the matrix entries can be described by a kernel function, which takes a "row" object, applies it to a "column" object, and outputs a number. We will show that, under mild assumptions, pairwise measurements are sufficient to create a vector space with dot product into which the "row" and "column" objects are mapped (cf. Section 2.6 and Appendix C for a theoretical investigation). We then suggest to construct the classifier or the regression function in analogy to the large margin based methods for learning perceptrons for vectorial data in this vector space. Using an improved measure for model complexity and a new set of constraints which ensure a good performance on the training data we arrive at a generally applicable method for learning predictors for matrix data. The new method, which we will call the potential support vector machine (P-SVM), can handle rectangular matrices as well as pairwise data whose matrices are not necessarily positive semidefinite. But even when the P-SVM is applied to regular Gram matrices, it shows excellent results when compared with standard methods. Due to the choice of constraints, the final predictor is expanded into a usually sparse set of descriptive "row" objects. This differs from standard support vector methods, where the predictor is expanded in terms of "column" objects. Expansion into "row" objects, however, opens up another important application domain: a sparse expansion is equivalent to feature selection (see Guyon and Elisseeff, 2003; Hochreiter and Obermayer, 2004a; Kohavi and John, 1997; Blum and Langley, 1997 for recent reviews on feature selection). In the following we will show, that the P-SVM can indeed be used in a prediction and in a feature selection mode, depending on the specific way the P-SVM is regularized. In the feature selection mode, the P-SVM extracts a small set of "row" objects which are particular informative (but not redundant) about the attributes which must be predicted. This improves the generalization performance of a subsequent prediction step, helps to understand the data generation process, and helps to identify the causes underlying the observed attributes.

In the following subsections, we first briefly review the classical support vector machine (SVM). Then we introduce a new scale-invariant objective function and present the new constraints which ensure a good performance on the training set. Then we suggest two different strategies for regularization in order to avoid overfitting for noisy data. The first strategy should be adopted when the P-SVM is used in prediction mode (classification and regression), the second strategy should be used, when the goal is feature selection. The performance

of both regularization schemes are illustrated using several toy data sets. Finally, the performance of the P-SVM is assessed using benchmarks with several real world data sets. The modified sequential minimal optimization procedure needed for learning is described in the appendix.

2 The Potential Support Vector Machine

2.1 Preliminaries

Consider a set $X = \left\{\boldsymbol{x}^i \mid 1 \leq i \leq L\right\}$ of L objects which are described by feature vectors $\boldsymbol{x}^i \in \mathbb{R}^N$. Consider for the moment a simple classification problem, where every object $\boldsymbol{x}^i$ belongs to one of two classes. Class membership is indicated by a binary label $y_i \in \{+1, -1\}$, and the labels for all objects in the training set are summarized by a label vector $\boldsymbol{y}$. Consider the set $\{\mathrm{sign}\,(f)\}$ of linear classifiers with

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

which are parameterized by the weight vector $\boldsymbol{w}$ and the offset b, where $\langle\cdot,\cdot\rangle$ denotes a dot product. The corresponding classification boundaries are given by the hyperplanes $f(\boldsymbol{x}) = 0$. The margin γ of a hyperplane with respect to the training set X, i.e. the distance between the hyperplane and the closest data point, is given by

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

If $\boldsymbol{w}$ and b are scaled, such that $\min_{\boldsymbol{x}\in X} |\langle \boldsymbol{w}, \boldsymbol{x}\rangle + b| = 1$ holds, then the hyperplane is in its "canonical form" (Vapnik, 1995) and we obtain $\gamma = \|\boldsymbol{w}\|_2^{-1}$.

Standard SVM-techniques select the hyperplane with the largest margin. They minimize $\|\boldsymbol{w}\|_2^2$ for all linear classifiers in their canonical form under the constraint of correct classification on the training set:

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

Clearly, the optimization problem eqs. (3) can only be solved if the training data are linearly separable. If not, a large margin has to be traded against a small training error using a suitable regularization scheme.

The maximum margin objective is motivated by bounds on the generalization error using the Vapnik-Chervonenkis (VC) dimension h as capacity measure (e.g. Vapnik, 1998). For the set of all linear classifiers defined on X, for which $\gamma \geq \gamma_{\min}$ holds, one obtains

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

(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. Minimizing model complexity h corresponds to maximizing the margin γ. Capacity measures and bounds derived using the fat shattering dimension (Shawe-Taylor et al., 1996, 1998; Schölkopf and Smola, 2002), and bounds on the *expected* generalization error (cf. Vapnik, 1998; Schölkopf and Smola, 2002) depend on $\frac{\mathrm{R}}{\gamma_{\min}}$ in a similar manner.

2.2 A Scale Invariant Objective Function

Both the selection of a classifier using the maximum margin principle and the values obtained for the bounds on the generalization error described in the last section suffer from the problem that they are not invariant under linear transformations. This problem is illustrated in Fig. 2. The figure shows a

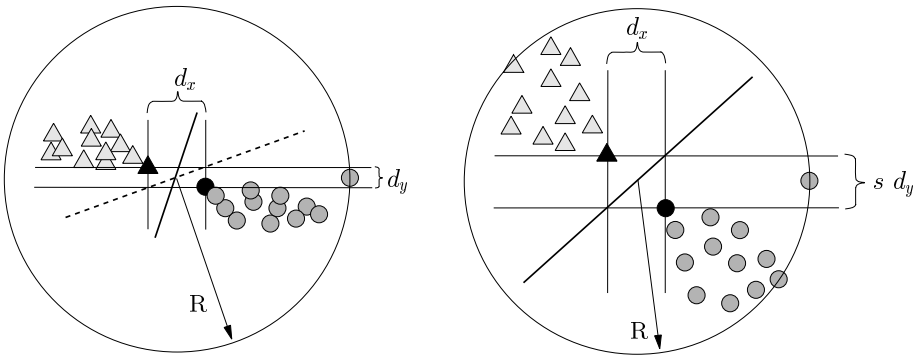

Figure 2: LEFT: data points from two classes (triangles and circles) are separated by the hyperplane with the largest margin (solid line). The two support vectors (black symbols) are separated by d_x along the horizontal and by d_y along the vertical axis, from which we obtain $\gamma = \frac{1}{2}\sqrt{d_x^2 + d_y^2}$ and $\frac{\mathrm{R}^2}{\gamma^2} = \frac{4\ \mathrm{R}^2}{d_x^2 + d_y^2}$. The dashed line indicates the classification boundary of the classifier shown on the right, scaled along the vertical axis by the factor $\frac{1}{s}$. RIGHT: the same data but scaled along the vertical axis by the factor s. The data points still lie within the sphere of radius R. The solid line denotes the maximum margin hyperplane. We obtain $\gamma = \frac{1}{2}\sqrt{d_x^2 + s^2\ d_y^2}$ and $\frac{\mathrm{R}^2}{\gamma^2} = \frac{4\ \mathrm{R}^2}{d_x^2 + s^2\ d_y^2}$. For $d_y \neq 0$ both the margin γ and the term $\frac{\mathrm{R}^2}{\gamma^2}$ depend on s.

two dimensional classification problem, where the data points from the two classes are indicated by triangles and circles. In the left figure, both classes are separated by the hyperplane with the largest margin (solid line). In the right figure, the same classification problem is shown, but scaled along the vertical axis by a factor s. Again, the solid line denotes the support vector solution,

but when the classifier is scaled back to $s = 1$ (dashed line in the left figure) it does no longer coincide with the original SVM solution. Therefore, the optimal hyperplane is not scale invariant and predictions of class labels may change if the data is rescaled before learning. In the legend of Fig. 2 it is shown that the ratio $\frac{\mathrm{R}^2}{\gamma^2}$, which bounds the VC dimension (see eq. (4)), also depends on the scale factor. Therefore, the question arises, which scale factors should be used for classifier selection.

Here we suggest to scale the training data such that the margin γ remains constant while the radius R of the sphere containing all training data becomes as small as possible. The result is a new sphere with radius $\tilde{\mathrm{R}}$ which still contains all training data but which leads to a tighter margin-based bound for the generalization error. Optimality is achieved when all directions orthogonal the normal vector $\boldsymbol{w}$ of the hyperplane with maximal margin γ are scaled to zero and $\tilde{\mathrm{R}} = \min_{t\in\mathbb{R}} \max_i |\langle \hat{\boldsymbol{w}}, \boldsymbol{x}^i \rangle + t| \leq \max_i |\langle \hat{\boldsymbol{w}}, \boldsymbol{x}^i \rangle|$, where $\hat{\boldsymbol{w}} := \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$. If the absolute value of t is small compared to the absolute values of $\langle \hat{\boldsymbol{w}}, \boldsymbol{x}^i \rangle$, e.g. if the data is centered around the origin, t can be neglected through above inequality. Unfortunately, above formulation does not lead to an optimization problem which is easy to implement. Therefore, we suggest to minimize the upper bound:

$$\begin{aligned} \frac{\tilde{\mathrm{R}}^2}{\gamma^2} &= \tilde{\mathrm{R}}^2 \, \|\boldsymbol{w}\|^2 \\ &\leq \max_i \langle \boldsymbol{w}, \boldsymbol{x}^i \rangle^2 \leq \sum_i \langle \boldsymbol{w}, \boldsymbol{x}^i \rangle^2 = \left\| \boldsymbol{X}^\top \boldsymbol{w} \right\|^2 , \end{aligned} \tag{5}$$

where we summarized training vectors $\boldsymbol{x}^i$ by the matrix $\boldsymbol{X} := \left(\boldsymbol{x}^1, \boldsymbol{x}^2, \ldots, \boldsymbol{x}^L\right)$.

It can be shown that replacing the objective function $\|\boldsymbol{w}\|^2$ in eqs. (3) by the upper bound

$$\boldsymbol{w}^\top \boldsymbol{X} \, \boldsymbol{X}^\top \boldsymbol{w} = \left\| \boldsymbol{X}^\top \boldsymbol{w} \right\|^2 \tag{6}$$

on $\frac{\tilde{\mathrm{R}}^2}{\gamma^2}$, eq. (5), corresponds to the integration of sphering (whitening) and SVM learning if the data have zero mean. Minimizing the new objective leads to normal vectors which tend to point in directions of low variance of the data. The new objective is well defined also for cases where $\boldsymbol{X} \, \boldsymbol{X}^\top$ or/and $\boldsymbol{X}^\top \, \boldsymbol{X}$ is singular, and the kernel trick carries over to the new technique. If the data has already been sphered, then the covariance matrix is given by $\boldsymbol{X} \, \boldsymbol{X}^\top = \mathbf{I}$ and we recover the classical SVM[2].

The new objective function, eq. (6), leads to separating hyperplanes which are invariant under linear transformations of the data. As a consequence, neither the bounds nor the performance of the derived classifier depends on how the training data was scaled. But is the new objective function also related to a

[2]In general, however, sphering as a preprocessing step does not simplify the problem because it must be based on kernel PCA if a kernel is used. Another disadvantage is that a tradeoff between sphering and low training error as discussed in Section 2.4 is no longer possible.

capacity measure for the model class like the margin is? It is, and in (Hochreiter and Obermayer, 2004b) it has been shown, that the capacity measure, eq. (6), emerges when a bound for the generalization error is constructed using the technique of covering numbers.

The new objective function of eq. (6) was motivated for a classification problem but it can also be used to find an optimal regression function in a regression problem. In regression the term $\left\|\boldsymbol{X}^\top \boldsymbol{w}\right\|^2 = \left\|\boldsymbol{X}^\top \hat{\boldsymbol{w}}\right\|^2 \|\boldsymbol{w}\|^2$, $\hat{\boldsymbol{w}} := \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$, is the product of a term which expresses the deviation of the data from the regression function and a term which corresponds to the smoothness of the regressor. If the regression function intersects the origin, which can be enforced by normalizing data vectors $\boldsymbol{x}$ to have zero mean (see eq. (23)) and by normalizing the attributes y_i such that $b = 0$ (see eq. (26)), then $\boldsymbol{X}^\top \hat{\boldsymbol{w}}$ is the vector of distances between the data and the regression function. The smoothness of the regression function is determined by the norm of the weight vector $\boldsymbol{w}$. For a simple linear function, $\|\boldsymbol{w}\|_2$ just determines the slope, but if nonlinear kernels are applied, the absolute values of the partial derivatives of the regression function are proportional to $\|\boldsymbol{w}\|_2$. Therefore, $\|\boldsymbol{w}\|_2$ determines the smoothness of the regression function. Let us assume that the nonlinear kernel k expresses a dot product in some feature space (see Section 2.6 and Appendix C for more details on kernels and dot products). Then a mapping $\Phi : \Phi(\boldsymbol{u}) = (\Phi_1(\boldsymbol{u}), \Phi_2(\boldsymbol{u}), \dots)$ into a feature space exists such that $k\left(\boldsymbol{u}^1, \boldsymbol{u}^2\right) = \left\langle \Phi(\boldsymbol{u}^1), \Phi(\boldsymbol{u}^2) \right\rangle$. We obtain for the regression function

$$f(\boldsymbol{u}) = \langle \boldsymbol{w}, \Phi(\boldsymbol{u}) \rangle + b \tag{7}$$

and for its gradient with respect to $\boldsymbol{u}$:

$$\nabla_{\boldsymbol{u}} f(\boldsymbol{u}) = \left(\nabla_{\boldsymbol{u}} \Phi(\boldsymbol{u})\right) \boldsymbol{w} \ , \tag{8}$$

where $\nabla_{\boldsymbol{u}} \Phi(\boldsymbol{u})$ is the matrix with the $\nabla_{\boldsymbol{u}} \Phi_i(\boldsymbol{u})$ as row vectors. The derivatives can be bounded using the Cauchy-Schwarz inequality:

$$\left\| \frac{\partial f(\boldsymbol{u})}{\partial u_j} \right\| = \left\| \left\langle \boldsymbol{w}, \frac{\partial \Phi(\boldsymbol{u})}{\partial u_j} \right\rangle \right\| \leq \|\boldsymbol{w}\| \left\| \frac{\partial \Phi(\boldsymbol{u})}{\partial u_j} \right\| \ , \tag{9}$$

where $\frac{\partial \Phi(\boldsymbol{u})}{\partial u_j}$ is the vector $\left(\frac{\partial \Phi_1(\boldsymbol{u})}{\partial u_j}, \frac{\partial \Phi_2(\boldsymbol{u})}{\partial u_j}, \dots \right)$. A smaller value of $\|\boldsymbol{w}\|$ leads to a smoother regression function. Minimizing eq. (6), therefore, leads to an optimal tradeoff between minimizing the distances between the data points and the regressor and maximizing the smoothness of the regression function. This tradeoff is reflected by eq. (94) in Appendix C which shows that eq. (6) is the L^2-norm of the function f.

2.3 New Constraints through Complex Features

We now consider the case, that the feature vectors $\boldsymbol{x}$ are not fully known. Instead we assume that a measurement device allows us to determine the values of a limited set of P complex features $\boldsymbol{z}$. The complex features $\boldsymbol{z}$ are linear combinations of the elementary features x_l, the components of the objects' feature

vectors $\boldsymbol{x}$, and they define directions in feature space. The value of a complex feature $\boldsymbol{z}^j$ for an object $\boldsymbol{x}^i$ is then given by the dot product

$$K_{ij} \;=\; \left\langle \boldsymbol{x}^i, \boldsymbol{z}^j \right\rangle \quad . \tag{10}$$

Let us summarize the different kinds of measurements using the complex feature matrix $\boldsymbol{Z} := \left(\boldsymbol{z}^1, \boldsymbol{z}^2, \ldots, \boldsymbol{z}^P\right)$. Then we can summerize our (incomplete) knowledge about the set of objects X using the data matrix $\boldsymbol{K}$,

$$\boldsymbol{K} \;=\; \boldsymbol{X}^\top \boldsymbol{Z} \quad . \tag{11}$$

In the case of DNA microarray data, for example, we could identify $\boldsymbol{K}$ with the matrix of expression values obtained by a microarray experiment. For web data we could identify $\boldsymbol{K}$ with the matrix of ingoing or outgoing hyperlinks. For documents we could identify $\boldsymbol{K}$ with the matrix of word frequencies. Hence we assume, that $\boldsymbol{x}$ and $\boldsymbol{z}$ live in a space of hidden causes which are responsible for the different attributes of the objects. The complex features $\left\{\boldsymbol{z}^j\right\}$ span a subspace of the original feature space, but we do not require them to be orthogonal, normalized, or linearly independent. If we set $\boldsymbol{z}^j \;=\; \boldsymbol{e}^j$ (jth Cartesian unit vector), that is $\boldsymbol{Z} \;=\; \boldsymbol{I}$, $K_{ij} \;=\; x_j^i$ and $P \;=\; N$, the "new" description, eq. (11), is fully equivalent to the "old" description using the original feature vectors $\boldsymbol{x}$.

We now turn to the task to define a quality measure for the performance of the classifier or the regression function on the training set. We consider the quadratic loss function

$$c(y_i, f(\boldsymbol{x}^i)) \;=\; \frac{1}{2} r_i^2 \;, \tag{12}$$

where

$$r_i \;=\; f(\boldsymbol{x}^i) - \; y_i \;=\; \left\langle \boldsymbol{w}, \boldsymbol{x}^i \right\rangle \;+\; b \;-\; y_i \tag{13}$$

is the residual error for a data point $\boldsymbol{x}^i$, defined as the difference between its attribute y_i and the value predicted by the classification or regression function f. The total residual error on the training set, the mean squared error, is

$$R_{\mathrm{emp}}\left[f_{\boldsymbol{w},b}\right] \;=\; \frac{1}{L} \sum_{i=1}^{L} c\left(y_i, f(\boldsymbol{x}^i)\right) \quad . \tag{14}$$

We now require, that the selected classification or regression function minimizes the total residual error, i.e. that

$$\nabla_{\boldsymbol{w}} R_{\mathrm{emp}}\left[f_{\boldsymbol{w},b}\right] \;=\; \frac{1}{L}\, \boldsymbol{X} \,\left(\boldsymbol{X}^\top \boldsymbol{w} \;+\; b\boldsymbol{1} \;-\; \boldsymbol{y}\right) \;=\; \boldsymbol{0} \tag{15}$$

and

$$\frac{\partial R_{\mathrm{emp}}[f]}{\partial b} \;=\; \frac{1}{L} \sum_i r_i \;=\; b \;+\; \frac{1}{L} \sum_i \left(\left\langle \boldsymbol{w}, \boldsymbol{x}^i \right\rangle \;-\; y_i\right) \;=\; 0 \;. \tag{16}$$

Since the quadratic loss function is convex in $\boldsymbol{w}$ and b, only one minimum exists if $\boldsymbol{X}\ \boldsymbol{X}^\top$ has full rank. If $\boldsymbol{X}\ \boldsymbol{X}^\top$ is singular, then all points of minimal value correspond to a subspace of $\mathbb{R}^N$. For the value of b we obtain from eq. (16)

$$b \ = \ -\frac{1}{L} \ \sum_{i=1}^{L} \left(\langle \boldsymbol{w}, \boldsymbol{x}^i \rangle \ - \ y_i \right) \ = \ -\frac{1}{L} \ \left(\boldsymbol{w}^\top \boldsymbol{X} - \boldsymbol{y}^\top \right) \mathbf{1} \ . \tag{17}$$

Condition eq. (15) implies, that the directional derivative should be zero along any direction in feature space, including the directions of the complex feature vectors $\boldsymbol{z}$. We, therefore, obtain

$$\begin{aligned} & \frac{dR_{\text{emp}} \left[f_{\boldsymbol{w} \ + \ t \ \boldsymbol{z}^j, b} \right]}{dt} \ = \ \left(\boldsymbol{z}^j \right)^\top \ \nabla_{\boldsymbol{w}} R_{\text{emp}} \left[f_{\boldsymbol{w}, b} \right] \\ & = \ \frac{1}{L} \ \left(\boldsymbol{z}^j \right)^\top \ \boldsymbol{X} \ \left(\boldsymbol{X}^\top \boldsymbol{w} \ + \ b \mathbf{1} \ - \ \boldsymbol{y} \right) \ = \ 0 \ , \end{aligned} \tag{18}$$

and, combining all complex features,

$$\begin{aligned} & \frac{1}{L} \ \boldsymbol{Z}^\top \ \boldsymbol{X} \ \left(\boldsymbol{X}^\top \boldsymbol{w} \ + \ b \mathbf{1} \ - \ \boldsymbol{y} \right) \ = \ \frac{1}{L} \ \boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} \ + \ b \mathbf{1} \ - \ \boldsymbol{y} \right) \\ & = \ \frac{1}{L} \ \boldsymbol{K}^\top \boldsymbol{r} \ = \ \mathbf{0} \ . \end{aligned} \tag{19}$$

Hence we require, that for every complex feature $\boldsymbol{z}^j$ the mixed moments σ_j between the residual error r_i and the measured values K_{ij} should be zero:

$$\begin{aligned} \sigma_j \ & = \frac{1}{L} \sum_{i=1}^{L} \ \left\langle \boldsymbol{x}^i, \boldsymbol{z}^j \right\rangle \ r_i \ = \ \frac{1}{L} \ \left[\boldsymbol{K}^\top \boldsymbol{r} \right]_j \\ & = \ \frac{dR_{\text{emp}} \left[f_{\boldsymbol{w} \ + \ t \ \boldsymbol{z}^j, b} \right]}{dt} \ = \ 0 \ . \end{aligned} \tag{20}$$

2.4 The Potential Support Vector Machine (P-SVM)

We now combine both the new objective from eq. (6) and the new constraints from eq. (19). The new procedure of selecting a classifier or a regression function is then given by

$$\begin{aligned} \min_{\boldsymbol{w}, b} \quad & \frac{1}{2} \ \| \boldsymbol{X}^\top \ \boldsymbol{w} \|^2 \\ \text{s.t.} \quad & \boldsymbol{K}^\top \ \left(\boldsymbol{X}^\top \ \boldsymbol{w} \ + \ b \mathbf{1} \ - \ \boldsymbol{y} \right) \ = \ \mathbf{0} \ . \end{aligned} \tag{21}$$

The number of constraints in this optimization problem is equal to the number P of complex features, which can be larger or smaller than the number L of data points or the dimension N of the original feature space. The constraints guarantee minimal mean squared error with respect to the complex features and can always be fulfilled because the minimum of the (convex) empirical error function fulfills the constraints (zero derivatives)[3]. f is chosen from all

[3]This fact is reflected through the projection of the residual errors $\boldsymbol{r} \ = \ \boldsymbol{X}^\top \ \boldsymbol{w} + b \mathbf{1} \ - \ \boldsymbol{y}$ onto the span of X via $\boldsymbol{K}^\top \boldsymbol{r} \ = \ \boldsymbol{Z}^\top \ \boldsymbol{X} \ \boldsymbol{r}$. Therefore, error components outside the span of X vanish and components inside the span of X can be forced to be zero for appropriate $\boldsymbol{w}$.

linear functions which are described by the P complex features and which have minimal mean squared error according to the objective function which measures f's capacity.

If $\boldsymbol{K}$ has at least rank L (number of training examples), then $\boldsymbol{r} = \mathbf{0}$ is always enforced. Consequently, if the measurements are noisy, overfitting occurs and the solutions to eqs. (21) are characterized by a high value of the objective function, eq. (6). Therefore, a regularization scheme is necessary, which allows for the violation of the constraints if a penalty is added to the objective function. In order to do so, the mixed moments σ_j must be normalized. The reason is, that high values of σ_j may either be a result of a high variance of the values of $\boldsymbol{K}_{ij}$ or the result of a high correlation between the residual error r_i and the values of K_{ij}. We are interested in the latter and want discard spurious correlations. The most appropriate measure would be Pearson's correlation coefficient

$$\hat{\sigma}_j \;=\; \frac{\sum_{i=1}^{L} \left(K_{ij} \;-\; \bar{K}_j\right) \; (r_i \;-\; r)}{\sqrt{\sum_{i=1}^{L} \left(K_{ij} \;-\; \bar{K}_j\right)^2} \sqrt{\sum_{i=1}^{L} \; (r_i \;-\; r)^2}} \quad , \tag{22}$$

where $r \;=\; \frac{1}{L} \; \sum_{i=1}^{L} \; r_i$ is the mean residual and $\bar{K}_j \;=\; \frac{1}{L} \; \sum_{i=1}^{L} \; K_{ij}$ is the mean value of the jth complex feature. If the data vectors $(K_{1j}, K_{2j}, \ldots, K_{Lj})$ are normalized to zero mean and unit variance,

$$\frac{1}{L} \sum_{i=1}^{L} \; \left(K_{ij} \;-\; \bar{K}_j\right)^2 \;=\; 1 \quad \text{and} \quad \bar{K}_j \;=\; \frac{1}{L} \; \sum_{i=1}^{L} \; K_{ij} \;=\; 0 \; , \tag{23}$$

we obtain

$$\sigma_j \;=\; \frac{1}{L} \sum_{i=1}^{L} \; K_{ij} \; r_i \;\; = \; \hat{\sigma}_j \; \frac{1}{\sqrt{L}} \; \|\boldsymbol{r} \;-\; r\mathbf{1}\|_2 \;\; . \tag{24}$$

The mixed moments are now proportional to the correlation coefficient $\hat{\sigma}_j$ with a proportionality constant which is independent of the complex feature $\boldsymbol{z}_j$. Note, that $r = 0$ is required by eq. (16). If eqs. (23) hold, σ_j can still be used instead of $\hat{\sigma}_j$ to formulate the constraints.

If the data vectors are normalized, the term $\boldsymbol{K}^\top \mathbf{1}$, which is is the factor in front of b in the constraints of problem eqs. (21), vanishes and we obtain the simplified optimization problem

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2} \; \|\boldsymbol{X}^\top \; \boldsymbol{w}\|^2 \\ \text{s.t.} \quad & \boldsymbol{K}^\top \; \left(\boldsymbol{X}^\top \; \boldsymbol{w} \;-\; \boldsymbol{y}\right) \;=\; \mathbf{0} \; , \end{aligned} \tag{25}$$

where the offset b of the classification or regression function is given by eq. (17). We will call this model selection procedure the **Potential Support Vector Machine (P-SVM)**, and we will always assume normalized data vectors in the following. Because of the normalization, eqs. (23), the equation for b, eq. (17), simplifies to

$$b \;=\; \frac{1}{L} \; \sum_{i=1}^{L} y_i \; . \tag{26}$$

The proof for this equation is provided in Appendix A.

In the next sections we show how the generic form of the P-SVM must be extended and that the regularization scheme essentially determines the application domain of the P-SVM. If slack variables are used, we obtain a machine which is useful for classification and regression. If a global threshold is used, the P-SVM is tailored for feature selection.

2.4.1 The P-SVM for Classification

If the P-SVM is used for classification, we suggest a regularization scheme based on slack variables $\boldsymbol{\xi}^+$ and $\boldsymbol{\xi}^-$. Slack variables allow for small violations of individual constraints if changing $\boldsymbol{w}$ would lead to a large increase of the objective function otherwise. We obtain

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{\xi}^+,\boldsymbol{\xi}^-} \quad & \frac{1}{2}\,\|\boldsymbol{X}^\top\,\boldsymbol{w}\|^2 \;+\; C\mathbf{1}^\top\left(\boldsymbol{\xi}^+ \;+\; \boldsymbol{\xi}^-\right) \\ \text{s.t.} \quad & \boldsymbol{K}^\top\;\left(\boldsymbol{X}^\top\,\boldsymbol{w} \;-\; \boldsymbol{y}\right) \;+\; \boldsymbol{\xi}^+ \;\geq\; \mathbf{0} \\ & \boldsymbol{K}^\top\;\left(\boldsymbol{X}^\top\,\boldsymbol{w} \;-\; \boldsymbol{y}\right) \;-\; \boldsymbol{\xi}^- \;\leq\; \mathbf{0} \\ & \mathbf{0} \;\leq\; \boldsymbol{\xi}^+, \boldsymbol{\xi}^- \end{aligned} \tag{27}$$

for the primal problem.

Above regularization scheme makes the optimization problem robust against "outliers". In general, a large value of the slack variables indicates, that the particular "row" object (complex feature) only weakly influences the direction of the classification boundary, because it would otherwise considerably increase the value of the complexity term. This happens in particular for high levels of measurement noise which leads to large, spurious values of the mixed moments σ_j. If the noise is large, the value of C must be small to "remove" the corresponding constraints via the slack variables $\boldsymbol{\xi}$. If the strength of the measurement noise is known, the correct value of C can be determined a priori. Otherwise, it takes the role of a hyperparameter and must be adapted using model selection techniques.

In order to derive the dual optimization problem, we have to evaluate the Lagrangian L,

$$\begin{aligned} L \;=\; & \frac{1}{2}\,\boldsymbol{w}^\top\,\boldsymbol{X}\,\boldsymbol{X}^\top\,\boldsymbol{w} \;+\; C\mathbf{1}^\top\left(\boldsymbol{\xi}^+ \;+\; \boldsymbol{\xi}^-\right) \qquad (28) \\ & -\; \left(\boldsymbol{\alpha}^+\right)^\top\;\left(\boldsymbol{K}^\top\left(\boldsymbol{X}^\top\,\boldsymbol{w} \;-\; \boldsymbol{y}\right) \;+\; \boldsymbol{\xi}^+\right) \\ & +\; \left(\boldsymbol{\alpha}^-\right)^\top\;\left(\boldsymbol{K}^\top\left(\boldsymbol{X}^\top\,\boldsymbol{w} \;-\; \boldsymbol{y}\right) \;-\; \boldsymbol{\xi}^-\right) \\ & -\; \left(\boldsymbol{\mu}^+\right)^\top\;\boldsymbol{\xi}^+ \;-\; \left(\boldsymbol{\mu}^-\right)^\top\;\boldsymbol{\xi}^- \;, \qquad (29) \end{aligned}$$

where the vectors $\boldsymbol{\alpha}^+ \geq \mathbf{0}$, $\boldsymbol{\alpha}^- \geq \mathbf{0}$, $\boldsymbol{\mu}^+ \geq \mathbf{0}$, and $\boldsymbol{\mu}^- \geq \mathbf{0}$ are the Lagrange multipliers for the constraints in eqs. (27). The optimality conditions (Schölkopf and Smola, 2002) require that

$$\begin{aligned} \nabla_{\boldsymbol{w}} L \;=\; & \boldsymbol{X}\,\boldsymbol{X}^\top\,\boldsymbol{w} \;-\; \boldsymbol{X}\,\boldsymbol{K}\,\boldsymbol{\alpha} \qquad (30) \\ =\; & \boldsymbol{X}\,\boldsymbol{X}^\top\,\boldsymbol{w} \;-\; \boldsymbol{X}\,\boldsymbol{X}^\top\boldsymbol{Z}\,\boldsymbol{\alpha} \;=\; \mathbf{0}\;, \end{aligned}$$

where we used the abbreviation $\boldsymbol{\alpha} = \boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-$ ($\alpha_i = \alpha_i^+ - \alpha_i^-$). In order to ensure eq. (30) and its equivalent equation $\boldsymbol{X}\,\boldsymbol{X}^\top\,\boldsymbol{w} = \boldsymbol{X}\,\boldsymbol{X}^\top\boldsymbol{Z}\,\boldsymbol{\alpha}$, we set

$$\boldsymbol{w} = \boldsymbol{Z}\,\boldsymbol{\alpha} \quad . \tag{31}$$

In contrast to the standard SVM expansion of $\boldsymbol{w}$ into its support vectors $\boldsymbol{x}$, the weight vector $\boldsymbol{w}$ is now expanded into a set of complex features $\boldsymbol{z}$ which we will call "support features" in the following. We then arrive at the dual optimization problem:

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^\top\,\boldsymbol{K}^\top\,\boldsymbol{K}\,\boldsymbol{\alpha} - \boldsymbol{y}^\top\,\boldsymbol{K}\,\boldsymbol{\alpha} \\ \text{s.t.} \quad & -C\,\mathbf{1} \le \boldsymbol{\alpha} \le C\mathbf{1} \quad . \end{aligned} \tag{32}$$

The dual problem is solved by a Sequential Minimal Optimization (SMO) technique which is described in Appendix B. The SMO technique is essential if many complex features are used, because in contrast to standard SVMs with a linear kernel it is the $N \times N$ correlation matrix $\boldsymbol{X}\,\boldsymbol{X}^\top$ and not the $L \times L$ Gram matrix $\boldsymbol{X}^\top\,\boldsymbol{X}$ which enters the dual formulation. Note, that for the $P \times P$ matrix $\boldsymbol{K}^\top\boldsymbol{K}$ we obtain $\boldsymbol{K}^\top\boldsymbol{K} = \boldsymbol{Z}^\top\,\boldsymbol{X}\,\boldsymbol{X}^\top\boldsymbol{Z}$.

Finally, the classification function f has to be constructed using the optimal values of the Lagrange parameters $\boldsymbol{\alpha}$. The value for b is given by eq. (26) and we obtain the classification function

$$\begin{aligned} f(\boldsymbol{u}) &= \langle \boldsymbol{w}, \boldsymbol{u} \rangle + b = \sum_{j=1}^{P} \alpha_j\,\langle \boldsymbol{u}, \boldsymbol{z}_j \rangle + b \\ &= \sum_{j=1}^{P} \alpha_j\,K_{(u)j} + b\ , \end{aligned} \tag{33}$$

where the expansion eq. (31) has been used for the weight vector $\boldsymbol{w}$.

The classifier based on eq. (33) depends on the weighting coefficients α_j, which were determined during optimization, on b, which can be computed directly by eq. (26), and on the measured values $\langle \boldsymbol{u}, \boldsymbol{z}^j \rangle$ for the new object $\boldsymbol{u}$. The weighting coefficients $\alpha_j = \alpha_j^+ - \alpha_j^-$ can be interpreted as class indicators, because they separate the complex features into features which are relevant for class 1 and class -1, according to the sign of α_j. Note, that if we consider the Lagrange parameters α_j as parameters of the classifier, we find that

$$\frac{dR_{\text{emp}}\left[f_{\boldsymbol{w}\,+\,t\,\boldsymbol{z}^j\,,\,b}\right]}{dt} = \sigma_j = \frac{\partial R_{\text{emp}}[f]}{\partial \alpha_j} \quad . \tag{34}$$

The directional derivatives of the empirical error R_{emp} along the complex features in the primal formulation correspond to its partial derivatives with respect to the corresponding Lagrange parameter in the dual formulation.

One of the most crucial properties of the P-SVM model selection procedure is, that the dual optimization problem only depends on $\boldsymbol{K}$ via $\boldsymbol{K}^\top\boldsymbol{K}$. Therefore, $\boldsymbol{K}$ is neither required to be positive semidefinite nor to be square. This allows

not only the construction of SVM-based classifiers for matrices $\boldsymbol{K}$ of general shape but also to extend the SVM-based approaches to the new class of indefinite kernels operating on the objects' feature vectors.

In the following we illustrate the application of the P-SVM approach to indefinite kernels using two toy examples. The first toy problem considers matrix data of the general form (see Fig. 1b). The data set consists of 34 "column" objects which are described by 2-dimensional feature vectors $\boldsymbol{x}$. 17 objects were chosen from class 1 and 17 objects from class 2 (see solid and open circles in Fig. 3). 50 "row" objects (complex features) and their 2-dimensional feature vectors $\boldsymbol{z}$ were chosen randomly according to an uniform distribution on the interval $[-2,2] \times [-2,2]$. We used three different types of kernels to calculate the matrix elements $K_{ij} = k\left(\boldsymbol{x}^i, \boldsymbol{z}^j\right)$: polynomial kernels, RBF-kernels, and the indefinite sine-kernel

$$k\left(\boldsymbol{x}^i, \boldsymbol{z}^j\right) \;=\; \sin\left(\omega \, \|\boldsymbol{x}^i \;-\; \boldsymbol{z}^j\|\right) \;.$$

The sine-kernel is indefinite, because its Gram matrix is zero in the main diagonal such that its trace vanishes. Since the trace of a matrix is the sum of its eigenvalues we deduce that the Gram matrix has both negative and positive eigenvalues (or is the zero matrix). Fig. 3 shows the results of the P-SVM method and demonstrates that good results can indeed be obtained with indefinite kernels.

The second toy problem considers pairwise data (see Fig. 1a). The data set consists of 70 objects, 28 from class 1 and 42 from class 2, which are described by two-dimensional feature vectors $\boldsymbol{x}$ (see open and solid circles in Fig. 4). A pairwise data set was then generated by applying the (indefinite) sine-kernel $k\left(\boldsymbol{x}^i, \boldsymbol{x}^j\right) \;=\; \sin\left(\omega \, \|\boldsymbol{x}^i - \boldsymbol{x}^j\|^2\right)$. In contrast to the previous example, we explicitly construct an indefinite Gram matrix using the non-Mercer sine-kernel. Fig. 4 shows the classification result obtained with the P-SVM method in comparison to the result using the standard RBF-kernel. The sine-kernel is more appropriate than the RBF-kernel for this data set which is indicated by the smaller number of support vectors. Note, that a large value of ω leads to a more "complex" set of classifiers (higher frequencies) and reduces the classification error on the training set. The figure demonstrates, that indefinite kernels cannot only be used to solve "standard" classification tasks, where objects are described by their feature vectors, but may lead to superior classification results due to the particular structure of the classification boundaries they induce (compare Fig. 4 left and right). The sine-kernel is clearly better adjusted to the "oscillatory" regions of class membership than the RBF-kernel is. This extends the range of kernels which are currently used and, therefore, opens up a new direction of research for kernel design.

2.4.2 The P-SVM for Regression

We have already argued at the end of Section 2.2, that the objective function, eq. (6), is also suitable for solving regression problems. The discussion in Section 2.3 and beginning of Section 2.4 also showed, that the constraints of vanishing

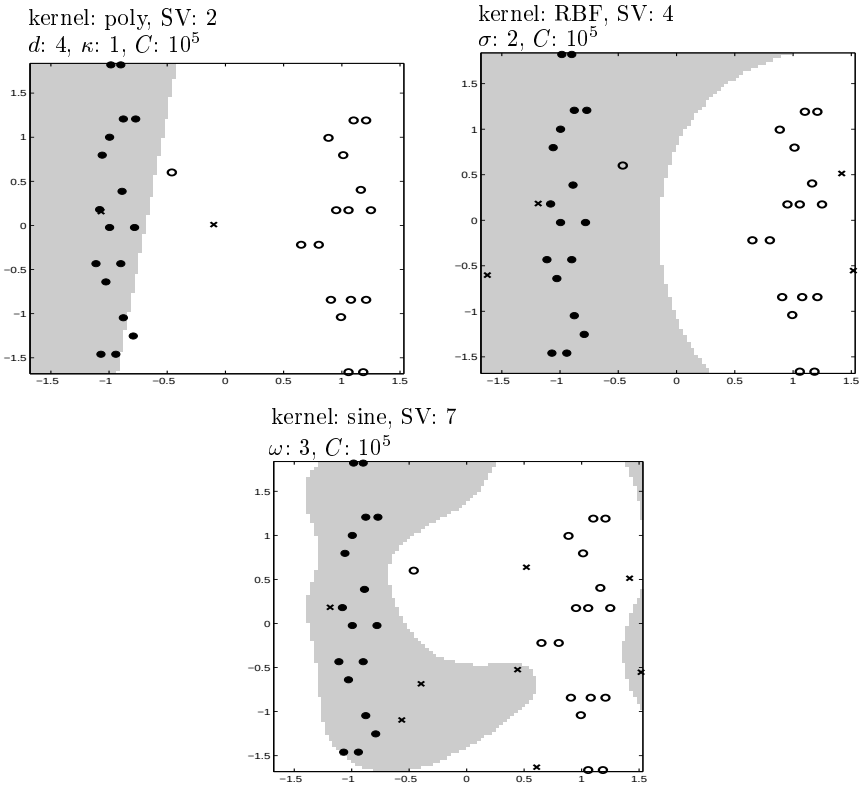

Figure 3: Application of the P-SVM method to a toy classification problem (matrix data). "Column" and "row" objects are described by two-dimensional feature vectors $\boldsymbol{x}$ and $\boldsymbol{z}$. 34 "column" objects, 17 from each class, were chosen as shown in the figure (open and solid circles), and 50 "row" objects (complex features) were generated randomly and uniformly from the interval $[-2, 2] \times [-2, 2]$. The figures show the resulting P-SVM classifier for the polynomial kernel $k\left(\boldsymbol{x}^i, \boldsymbol{z}^j\right) = \left(\left\langle \boldsymbol{x}^i, \boldsymbol{z}^j \right\rangle + \kappa\right)^d$ (poly), the RBF kernel $k\left(\boldsymbol{x}^i, \boldsymbol{z}^j\right) = \exp(-\frac{1}{\sigma^2} \left\|\boldsymbol{x}^i - \boldsymbol{z}^j\right\|^2$ (RBF), and the sine-kernel $k\left(\boldsymbol{x}^i, \boldsymbol{z}^j\right) = \sin(\omega \left\|\boldsymbol{x}^i - \boldsymbol{z}^j\right\|)$ (sine). Gray and white regions indicate areas of class 1 and class 2 membership as predicted by the selected classifiers and crosses indicate support features. Parameters are given in the figure.

mixed moments carry over to regression problems with the only modification, that the target values y_i in eqs. (27) are real rather than binary (± 1) numbers. The constraints are even more "natural" for regression because the r_i are indeed the residuals a regression function should minimize. We, therefore, propose to use the primal optimization problem, eqs. (27), and its corresponding dual, eqs. (32), also for the regression setting.

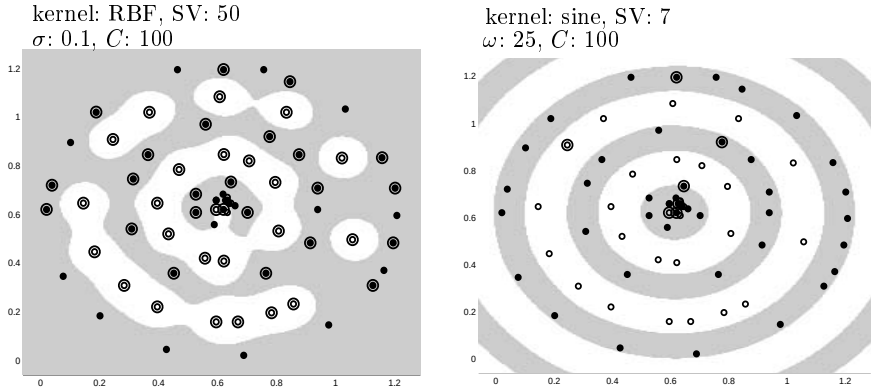

Figure 4: Application of the P-SVM method to a toy classification problem (pairwise data). Objects are described by two-dimensional feature vectors $\boldsymbol{x}$, and 70 objects were generated of which 28 belong to class 1 (open circles) and 42 belong to class 2 (solid circles). A Gram matrix was constructed using the positive definite RBF kernel (left) and the indefinite sine-kernel $k\left(\boldsymbol{x}^i, \boldsymbol{x}^j\right) = \sin\left(\omega \left\|\boldsymbol{x}^i - \boldsymbol{x}^j\right\|\right)$ (right). White and gray indicate regions of class 1 and class 2 membership. Circled data indicate support vectors. Parameters are given in the figure.

Fig. 5 shows the application of the P-SVM to a toy regression example (pairwise data). 50 data points were randomly chosen from the true function (dashed line) and i.i.d. Gaussian noise with mean 0 and standard deviation 0.2 were added to each y-component. One outlier was added by hand at $x = 0$. The figure shows the P-SVM regression results (solid lines) for an RBF-kernel and three different combinations of C and σ. It can be seen that the hyperparameter C controls the sensitivity against outliers, hence the local smoothness of the regressor. A smaller value of C reduces the height of the peak at $x = 0$ and the effect of this particular data point. However, smaller values of C also lead to a larger number of support vectors. The width σ of the RBF-kernel on the other hand controls the overall smoothness of the regressor. It reduces the influence of outliers without increasing the number of support vectors but at the expense of a large training error, that is large bias. The effect of local vs. global smoothing can be seen at $x = -2$ (cf. arrows in Fig. 5): σ-smoothing (upper left sub-figure in Fig. 5) results in a regression function with an increased training error contribution at $x = -2$ and no support vectors whereas C-smoothing (lower sub-figure in Fig. 5) leads to a lower training error contribution at $x = -2$ but more support vectors.

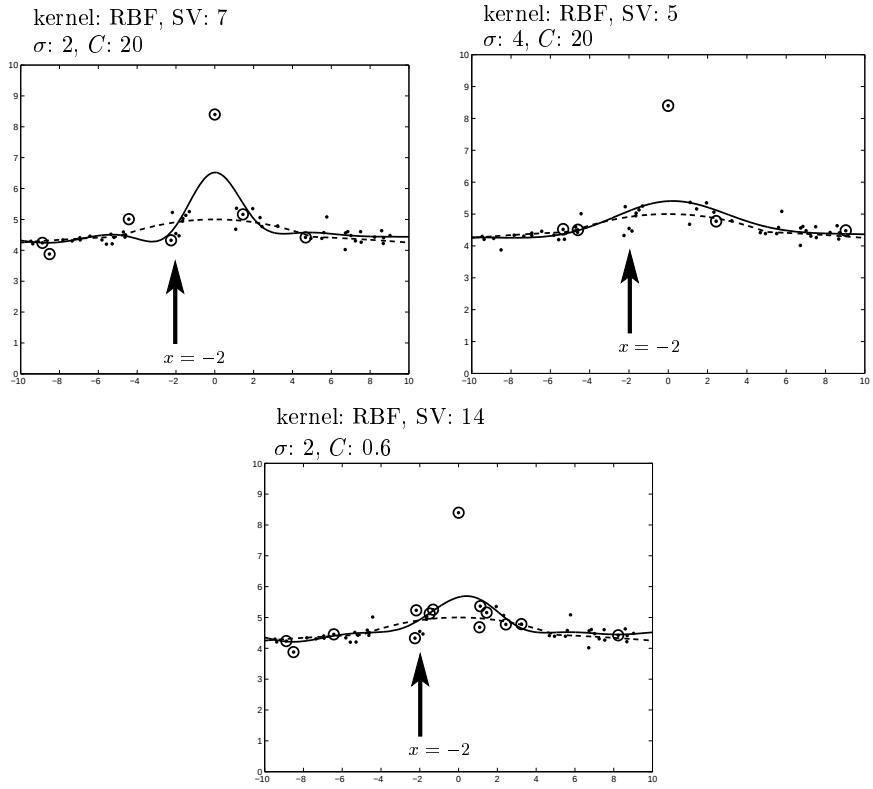

Figure 5: Application of the P-SVM method to a toy regression problem (pairwise data). Objects (small dots), described by the x-coordinate, were generated by randomly choosing points from the true function (dashed line) and adding Gaussian noise with mean 0 and standard deviation 0.2 to the y-component of each data point. One outlier was added by hand at $x = 0$. A Gram matrix was then generated using an RBF-kernel with width σ. The solid lines show the regression result. Circled dots indicate support vectors. Parameters are given in the figure. The arrows in the figures mark $x = -2$, where the effect of local vs. global smoothing can be seen (see text for explanation).

2.4.3 The P-SVM for Feature Selection

In this section we modify the P-SVM method for feature selection such that it can serve as a data preprocessing method in order to improve the generalization performance of subsequent classification or regression tasks (see also Hochreiter and Obermayer, 2004a). Due to the property of the P-SVM method to expand $\boldsymbol{w}$ into a sparse set of support features, it can be modified to optimally extract a small set of "informative" features with respect to certain attributes of the "column" objects (class labels or real valued attributes). The most important

modification is to adopt a different regularization scheme, which we will detail below. The set of "support features" can then be identified with the set of "selected" features, which may then be used as input to an arbitrary predictor, e.g. a standard SVM or a K-nearest-neighbor classifier.

Noisy measurements can lead to spurious mixed moments, i.e. complex features which contain no information about the objects' attributes but still exhibit finite values of σ_j. In order to prevent those features to affect the classification boundary or the regression function, we introduce a "correlation threshold" ϵ and modify the constraints in problem eqs. (25) according to

$$\| \boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} - \boldsymbol{y} \right) \|_\infty \leq \epsilon \,, \tag{35}$$

which can be written as

$$\begin{aligned} \boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} - \boldsymbol{y} \right) - \epsilon \, \mathbf{1} &\leq \mathbf{0} \,, \\ \boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} - \boldsymbol{y} \right) + \epsilon \, \mathbf{1} &\geq \mathbf{0} \;. \end{aligned} \tag{36}$$

This regularization scheme is analogous to the ϵ-insensitive loss (Schölkopf and Smola, 2002). Absolute values of mixed moments smaller than ϵ are considered to be spurious. Consequently, the influence of the corresponding features do not influence the weight vector, because the constraints remain fulfilled.

Similar to the classification case, high levels of noise induce stronger spurious correlations and the value of ϵ must be increased. The measurement noise directly correlates with ϵ, hence ϵ can be determined a priori if the level of measurement noise is known. If the level of noise level is unknown, ϵ serves as hyperparameter and its value can be determined using model selection techniques. Note, that data vectors have to be normalized (cf. eqs. (23)) before applying the P-SVM, because otherwise a global value of ϵ would not suffice.

Combining eq. (6) and eqs. (37) we then obtain the primal optimization problem

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2} \, \| \boldsymbol{X}^\top \boldsymbol{w} \|^2 \\ \text{s.t.} \quad & \boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} - \boldsymbol{y} \right) + \epsilon \, \mathbf{1} \geq \mathbf{0} \\ & \boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} - \boldsymbol{y} \right) - \epsilon \, \mathbf{1} \leq \mathbf{0} \end{aligned} \tag{37}$$

for P-SVM feature selection. In order to derive the dual formulation we have to evaluate the Lagrangian:

$$\begin{aligned} L \;=\; & \frac{1}{2} \, \boldsymbol{w}^\top \, \boldsymbol{X} \, \boldsymbol{X}^\top \, \boldsymbol{w} \\ & - \left(\boldsymbol{\alpha}^+ \right)^\top \left(\boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} - \boldsymbol{y} \right) + \epsilon \, \mathbf{1} \right) \\ & + \left(\boldsymbol{\alpha}^- \right)^\top \left(\boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} - \boldsymbol{y} \right) - \epsilon \, \mathbf{1} \right) \quad , \end{aligned} \tag{38}$$

where we have used the notation from Section 2.4.1. The vector $\boldsymbol{w}$ is again expressed through the complex features,

$$\boldsymbol{w} \;=\; \boldsymbol{Z} \, \boldsymbol{\alpha} \;, \tag{39}$$

and we obtain the dual formulation of eq. (38):

$$\begin{aligned} \min_{\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-} \quad & \frac{1}{2}\left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right)^\top \boldsymbol{K}^\top \boldsymbol{K} \left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right) \\ & - \boldsymbol{y}^\top \boldsymbol{K} \left(\boldsymbol{\alpha}^+ - \boldsymbol{\alpha}^-\right) + \epsilon\, \mathbf{1}^\top \left(\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-\right) \\ \text{s.t.} \quad & \mathbf{0} \leq \boldsymbol{\alpha}^+ \,, \quad \mathbf{0} \leq \boldsymbol{\alpha}^- \,. \end{aligned} \tag{40}$$

The term $\epsilon\, \mathbf{1}^\top (\boldsymbol{\alpha}^+ + \boldsymbol{\alpha}^-)$ in this dual objective function enforces a sparse expansion of the weight vector $\boldsymbol{w}$ in terms of the support features. This occurs, because for large enough values of ϵ, this term forces all α_j towards zero except for the complex features which are most relevant for classification or regression. If $\boldsymbol{K}^\top \boldsymbol{K}$ is singular and $\boldsymbol{w}$ is not uniquely determined, ϵ enforces a unique solution, which is characterized by the most sparse representation through complex features.

The dual problem is again solved by a Sequential Minimal Optimization (SMO) technique (see Appendix B). A fast SMO technique is crucial because in typical feature selection problems the number of features can be extremely large, hence the optimization problem, which is quadratic in the number of complex features, may become computationally very expensive.

Finally, let us address the relationship between the value of a Lagrange multiplier α_j and the "importance" of the corresponding complex feature $\boldsymbol{z}^j$ for prediction. The change of the empirical error under a change of the weight vectors by an amount β along the direction of a complex feature $\boldsymbol{z}^j$ is given by

$$\begin{aligned} & R_{\text{emp}}\left[f_{\boldsymbol{w}\,+\,\beta\,\boldsymbol{z}^j\,,\,b}\right] - R_{\text{emp}}\left[f_{\boldsymbol{w},b}\right] \\ & = \beta\,\sigma_j + \frac{\beta^2}{2\,L}\sum_i K_{ij}^2 = \beta\,\sigma_j + \frac{\beta^2}{2} \\ & \leq \frac{\epsilon\,|\beta|}{L} + \frac{\beta^2}{2}\,, \end{aligned} \tag{41}$$

because the constraints eq. (37) ensure that $|\sigma_j|\, L \leq \epsilon$. If a complex feature $\boldsymbol{z}^j$ is completely removed, then $\beta = -\alpha_j$ and

$$R_{\text{emp}}\left[f_{\boldsymbol{w}\,-\,\alpha_j\,\boldsymbol{z}^j\,,\,b}\right] - R_{\text{emp}}\left[f_{\boldsymbol{w},b}\right] \leq \frac{\epsilon\,|\alpha_j|}{L} + \frac{\alpha_j^2}{2}\,. \tag{42}$$

The Lagrange parameter α_j is directly related to the increase in the empirical error. Therefore, $\boldsymbol{\alpha}$ serve as importance measures for the complex features.

In the following, we illustrate the application of the P-SVM approach to feature selection using two toy examples. The first toy example considers matrix data in the general form (Fig. 1b) and a classification task. The data set consists of 50 "column" objects, 25 from each class, which are described by two-dimensional feature vectors $\boldsymbol{x}$ (open and solid circles in Fig. 6). 50 "row" objects and their two-dimensional feature vectors $\boldsymbol{z}$ were chosen randomly according to an uniform distribution on the interval $[-1.2, 1.2] \times [-1.2, 1.2]$. The data matrix $\boldsymbol{K}$ was generated using an RBF kernel with variance $\sigma = 0.2$. Fig. 6 shows the result of the P-SVM feature selection method with a correlation

threshold $\epsilon = 20$. The selected features are indicated by crosses. The figure shows, that every group of data points is described (and detected) by one or two feature vectors.

The six features selected by the P-SVM method are sufficient to allow for an almost perfect classification of new data points if they are chosen from the same distribution. The number of selected features depends on σ, which determines how the strength of correlation between complex features and objects decrease with their distance. Note, that for the RBF-kernel we obtain

$$k\left(\boldsymbol{x}^i, \boldsymbol{z}^j\right) \;=\; K^{\phi}_{i,j} \;=\; \left\langle \phi\left(\boldsymbol{x}^i\right), \phi\left(\boldsymbol{z}^j\right)\right\rangle \;=\; \exp\left(-\frac{1}{2\,\sigma^2}\,\|\boldsymbol{x}^i \;-\; \boldsymbol{z}^j\|^2\right)$$

which means that smaller distances $\|\boldsymbol{x}^i \;-\; \boldsymbol{z}^j\|$ in the input space indicate higher dot product values $K^{\phi}_{i,j}$ in feature space. The threshold ϵ determines the minimum value of non-spurious correlation. Smaller ϵ or larger σ would result in more complex features assigned to every data group.

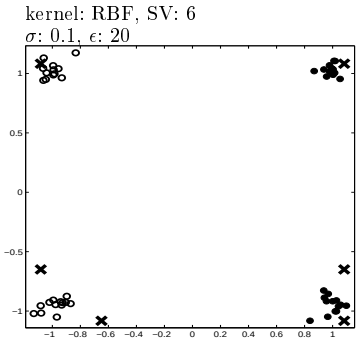

Figure 6: Application of the P-SVM method to a toy feature selection problem for a classification task (matrix data). "Column" and "row" objects are described by two-dimensional feature vectors $\boldsymbol{x}$ and $\boldsymbol{z}$, respectively. 50 "column" objects, 25 from each class (open and solid circles), were generated by randomly choosing a center from $\{(1,1),(1,-1),(-1,1),(-1,-1)\}$ with equal probability, then adding to each coordinate of the center a random value, which stems from a Gaussian with mean 0 and standard deviation 0.1. 100 "row" objects (complex features) were generated randomly and uniformly from the interval $[-1.2, 1.2] \times [-1.2, 1.2]$. An RBF-kernel $\exp\left(-\frac{1}{2\,\sigma^2}\,\|\boldsymbol{x}^i \;-\; \boldsymbol{z}^j\|^2\right)$ with width $\sigma = 0.2$ is applied to each pair $\left(\boldsymbol{x}^i, \boldsymbol{z}^j\right)$ of "row" and "column" object in order to construct the data matrix $\boldsymbol{K}$. Black crosses indicate the location of features selected by the P-SVM method.

The second toy problem considers feature selection for pairwise data (Fig. 1a) in the context of a regression task. The data set consists of 100 data points which are randomly chosen from the true function (dashed line) and for which Gaussian noise with mean 0 and standard deviation 10 was added to the y-components. A

Gram matrix was constructed using an RBF-kernel with width $\sigma = 2$, and the P-SVM method was applied for feature selection. Fig. 7 shows the regression function (solid line) and the selected support features (circled dots) for different values of the correlation threshold ϵ. The number of selected features decreases with increasing values for ϵ as expected. Interestingly, the support vectors indicate maxima and minima of the regression function (cf. bottom of Fig. 7), that means the support vectors mark the most interesting regions of the regression function. A more detailed regression function is obtained by more support vectors, however, they are broader distributed over the input space of the regression function (cf. top left of Fig. 7).

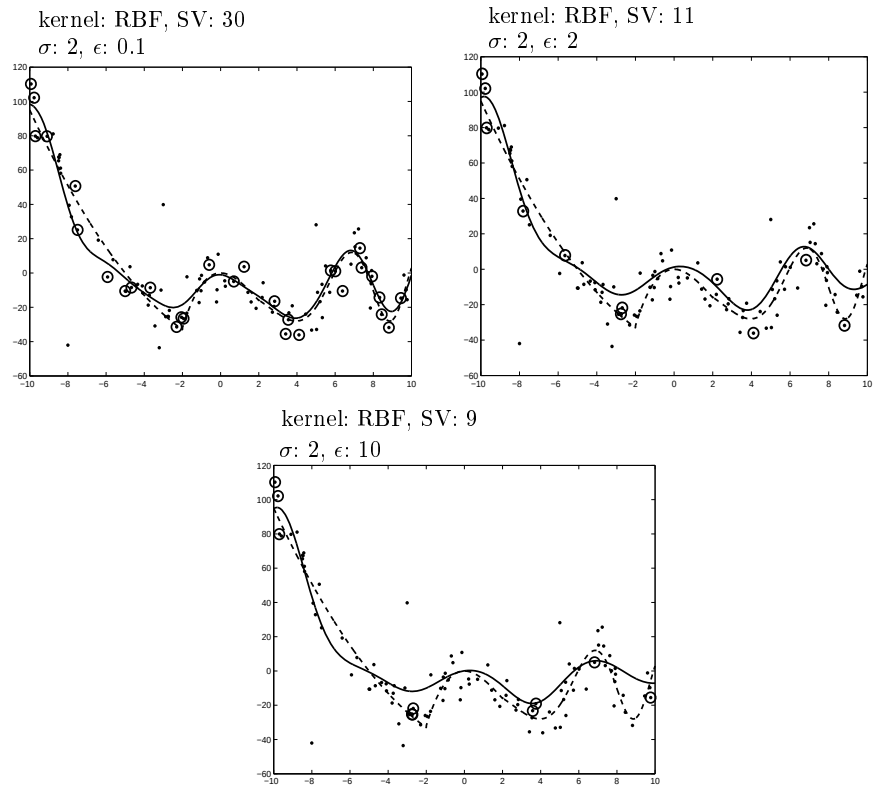

Figure 7: Application of the P-SVM to a toy feature selection problem for a regression task (pairwise data). 100 data points are generated from the true function (dashed line) by randomly and uniformly choosing data points from the true function and adding Gaussian noise with mean 0 and standard deviation 10 to the function value. A Gram matrix was constructed using an RBF-kernel with width $\sigma = 2$. The figure shows the P-SVM regression functions (solid lines) and the selected support vectors (circled dots). Parameters are given in the figure.

2.5 Duality Between the Two Regularization Schemes

We now directly compare the two regularization schemes proposed for classification and regression (slack variables) and feature selection (correlation threshold). The slack variables $\boldsymbol{\xi}$ can be eliminated from the optimization problem of eqs. (27) and we obtain

$$\min_{\boldsymbol{w}} \quad \frac{1}{2} \, \|\boldsymbol{X}^\top \, \boldsymbol{w}\|^2 \; + \; C \, \|\boldsymbol{K}^\top \, \left(\boldsymbol{X}^\top \, \boldsymbol{w} \; - \; \boldsymbol{y}\right) \|_1 \tag{43}$$

for the primal and

$$\begin{aligned} \min_{\boldsymbol{\alpha}} \quad & \frac{1}{2} \boldsymbol{\alpha}^\top \, \boldsymbol{K}^\top \, \boldsymbol{K} \, \boldsymbol{\alpha} \; - \; \boldsymbol{y}^\top \, \boldsymbol{K} \, \boldsymbol{\alpha} \\ \text{s.t.} \quad & \|\boldsymbol{\alpha}\|_\infty \; \leq \; C \end{aligned} \tag{44}$$

for the dual formulation (cf. eqs. (32)). The slack variables can be eliminated because (i) for each j, either $\xi_j^+ \; = \; 0$ or $\xi_j^- \; = \; 0$ and (ii) the remaining non-zero $\xi_j^\pm$ are equal to the absolute value $\left| \left[\boldsymbol{K}^\top \; \left(\boldsymbol{X}^\top \, \boldsymbol{w} \; - \; \boldsymbol{y} \right) \right]_j \right|$.

For the optimization problem of eqs. (38), the "correlation threshold" regularization, we obtain

$$\begin{aligned} \min_{\boldsymbol{w}} \quad & \frac{1}{2} \, \|\boldsymbol{X}^\top \, \boldsymbol{w}\|^2 \\ \text{s.t.} \quad & \|\boldsymbol{K}^\top \, \left(\boldsymbol{X}^\top \, \boldsymbol{w} \; - \; \boldsymbol{y}\right) \|_\infty \; \leq \; \epsilon \; . \end{aligned} \tag{45}$$

This leads to the dual formulation

$$\min_{\boldsymbol{\alpha}} \quad \frac{1}{2} \boldsymbol{\alpha}^\top \, \boldsymbol{K}^\top \, \boldsymbol{K} \, \boldsymbol{\alpha} \; - \; \boldsymbol{y}^\top \, \boldsymbol{K} \, \boldsymbol{\alpha} \; + \; \epsilon \, \|\boldsymbol{\alpha}\|_1 \; . \tag{46}$$

This dual is derived from the feature selection dual eqs. (41), where either $\alpha_j^+ \; = \; 0$ or $\alpha_j^- \; = \; 0$. If this would not be the case the term $\epsilon \; \mathbf{1}^\top \left(\boldsymbol{\alpha}^+ \; + \; \boldsymbol{\alpha}^-\right)$ can be decreased without changing the other terms in eqs. (41) by simply subtracting $\min \left\{\alpha_i^+, \alpha_i^-\right\}$ from α_i^+ and α_i^-. Thus, for $\boldsymbol{\alpha} \; = \; \boldsymbol{\alpha}^+ \; - \; \boldsymbol{\alpha}^-$ we obtain $\epsilon \; \mathbf{1}^\top \left(\boldsymbol{\alpha}^+ \; + \; \boldsymbol{\alpha}^-\right) \; = \; \epsilon \, \|\boldsymbol{\alpha}\|_1$ and above dual.

A direct comparison of the optimization problems shows that exchanging the "slack variables" and the "correlation threshold" regularization schemes result in simply exchanging the ∞-norm with the 1-norm for the dual variables $\boldsymbol{\alpha}$ and the primal constraints:

$$\begin{aligned} &\text{class./regression:} \quad \|\boldsymbol{\alpha}\|_\infty \; \leq \; C \; , \quad \min \; \|\boldsymbol{K}^\top \; \left(\boldsymbol{X}^\top \, \boldsymbol{w} \; - \; \boldsymbol{y}\right) \|_1 \\ &\text{feature selection:} \quad \min \; \|\boldsymbol{\alpha}\|_1 \; , \quad \|\boldsymbol{K}^\top \; \left(\boldsymbol{X}^\top \, \boldsymbol{w} \; - \; \boldsymbol{y}\right) \|_\infty \; \leq \; \epsilon \; . \end{aligned}$$

According to the Karush-Kuhn-Tucker conditions either the dual variable or the corresponding constraint must be zero. The two regularization schemes differ by which of the two factors is pushed towards zero: For the "correlation threshold" regularization the vector $\boldsymbol{\alpha}$ is made sparse leading to an optimal small set of support features while for the "slack variable" regularization scheme the vector of constraints is made sparse leading to low classification or regression costs

on the training set. This explains why the "slack variable" and "correlation threshold" schemes are particularly well suited for classification / regression and feature selection tasks, respectively.

Note that only the problem eq. (47) always yields a unique solution, i.e. the SMO algorithm for feature selection has a well defined convergence criterion. For all other optimization problems the solutions $\boldsymbol{w}$ ($\boldsymbol{\alpha}$) can have arbitrary parts in the subspace of matrix $\boldsymbol{X}$ ($\boldsymbol{K}$), which is mapped to zero. Therefore, we used for practical reasons also for classification and regression experiments ("slack variable" scheme) a small ϵ to enforce a stable solution of the dual optimization problem.

2.6 Matrix and Pairwise Data as Dot Product

In the derivation of the P-SVM method we have used the fact that the matrix $\boldsymbol{K}$ is a dot product matrix whose elements denote a scalar product between the feature vectors which describe the "row" and the "column" objects. If $\boldsymbol{K}$, however, is a matrix of measured values the question arises under which conditions such a matrix can be interpreted as a dot product matrix.

There is no full answer to this question from a theoretical viewpoint, practical applications have to confirm (or disprove) the chosen ansatz and data model. However, the question whether it is possible to describe a measurement operator which takes a "row" and a "column" object and outputs a number by a dot product can be replaced by the question whether or not the following three conditions hold:

(1) "Column" objects ("samples") $\boldsymbol{x}$ are from a set $\mathcal{X}$ which can be completed to a measure space.

(2) "Row" objects ("complex features") $\boldsymbol{z}$ are from a set $\mathcal{Z}$ which can be completed to a measure space.

(3) The measurement process can be expressed via the evaluation of a measurable kernel $k(\boldsymbol{x}, \boldsymbol{z})$ which is from $L^2(\mathcal{X}, \mathcal{Z})$.

In general, conditions (1) and (2) can easily fulfilled by defining a suited σ-algebra on the sets. Condition (3) holds for bounded k and compact sets $\mathcal{X}$ and $\mathcal{Z}$, i.e. for measurements where the measured value is bounded. These are mild conditions on the measurement.

Condition (3) equates the evaluation of a kernel as known from standard SVMs with physical measurements. As the kernel matrix is measured, no model selection has to be performed w.r.t. the kernel. The physical characteristics of the measurement device determines the properties of the kernel, e.g. boundedness and continuity. The theoretical analysis for the connection between kernels and dot products is provided in Appendix C, where we also show that indefinite kernels in the context of pairwise data correspond to dot products in Minkowski spaces, i.e. in a linear space equipped with an indefinite norm. This analysis

for pairwise data is based on the fact that $\boldsymbol{K}$ and the underlying kernel is symmetric. The most important fact is that neither the set of "column" objects $\mathcal{X}$ nor the set of "row" objects $\mathcal{Z}$ must be vector spaces. Therefore, objects can be classified if they can be related to other objects even if they do not allow a vectorial representation.

We derive additional interesting facts in Appendix C:

(1) The space in which the measurement kernel evaluates a dot product can be identified with ℓ^2, the space of infinite vectors with finite Euclidean length.

(2) For the discrete case we obtain $\|f\|_{L^2}^2 = \boldsymbol{\alpha}^\top \boldsymbol{K}^\top \boldsymbol{K}\ \boldsymbol{\alpha} = \|\boldsymbol{X}^\top\ \boldsymbol{w}\|_2^2$. Therefore the new objective eq. (6) is the L^2 norm of the classifier. This again suggests to use the new objective function eq. (6) as a capacity measure.

3 Numerical Experiments and Applications

In this section we apply the P-SVM method to various kinds of real world data sets and provide benchmark results with previously proposed methods when appropriate. This section consists of three parts which cover classification, regression, and feature selection. In part one the P-SVM is first tested as a classifier on data sets from the UCI Benchmark Repository and its performance is compared with results obtained for the C- and the ν-SVMs for different kernels. Then we apply the P-SVM to two measured (rather than constructed) pairwise data sets ("cat cortex" and "protein") and one measured matrix data set ("World Wide Web"). In part two the P-SVM is applied to regression problems taken from the UCI Benchmark Repository and compared to results obtained with C-Support Vector Regression and Bayesian SVMs. Part three describes results obtained for the P-SVM as a feature selection method for the "protein" and "World Wide Web" data sets, for a challenging toy data set similar to (Weston et al., 2000), and several real world data sets obtained using the DNA microarray technique (Pomeroy et al., 2002; Shipp et al., 2002; van't Veer et al., 2002). For the toy data set the performance of the P-SVM is compared with several standard feature selection methods: Fisher statistics (Kendall and Stuart, 1977), Recursive Feature Elimination (RFE) (Guyon et al., 2002), and R2W2 (Weston et al., 2000). Since no ground truth is available for the indicative features in the real world data sets, the feature selection methods are evaluated with respect to the performance of an optimal classifier operating on the selected set. Benchmarks are provided which compare the P-SVM / ν-SVM method and the methods "known most important gene" / one gene classification SPLASH / likelihood ratio classifier signal-to-noise-statistics / K-nearest neighbor, signal-to-noise-statistics / weighted voting, Fisher statistics / weighted voting, and R2W2, whose results are reported in the literature (Pomeroy et al., 2002; Shipp et al., 2002; van't Veer et al., 2002). SPLASH is a greedy subset selection method (Califano et al., 1999), likelihood ratio classifier uses a density estimation for

each feature and class, and weighted voting is linear classifier which multiplies the feature values by their statistical significance. All methods are described in more detail in the according literature.

3.1 Application to Classification Problems

3.1.1 UCI Data Sets

In this section we report benchmark results for the data sets "thyroid" (5 features), "heart" (13 features), "breast-cancer" (9 features), and "german" (20 features) from the UCI benchmark repository, and for the data set "banana" (2 features) taken from (Rätsch et al., 2001). All data sets were preprocessed as described in (Rätsch et al., 2001) and divided into 100 training/test set pairs. Data sets were generated 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 every data set we restricted the training set to the first 200 examples of the original training data set because otherwise the classification problem was too simple and the results did not differ significantly for the different methods. For testing we used the original test sets. Pairwise datasets were generated by constructing the Gram matrix for radial basis function (RBF), polynomial (POL), and Plummer (PLU, see Hochreiter et al., 2003) kernels, and the Gram matrices were used as input for C-, ν-, and P-SVM. Hyperparameters (C, ν, and kernel parameters) were optimized using 5–fold cross validation on the corresponding training sets. To ensure a fair comparison, the hyperparameter selection procedure was equal for all methods, but the search for hyperparameter was not as exhaustive as in (Rätsch et al., 2001).

Table 1 summarizes the percentage of misclassification averaged over 100 experiments. Despite the fact that C– and ν–SVMs are equivalent, results differ because of different model selection with hyperparameters C and ν. Best and second best results are indicated by bold and italic numbers, and a total score was calculated for every method by adding 2 points if it has won and 1 point if it was second best. The C-SVM was one times best and one times second best, the ν-SVM was two times best, and the P-SVM was three times best and four times second best. Therefore the score is 3 points for the C-SVM, 4 points for the ν-SVM, and 10 points for the P-SVM. The results show that the P-SVM method achieves the best result.

3.1.2 Cat Cortex Data Set

The "cat cortex" data set was taken from (Scannell et al., 1995) and describes the connectivity pattern between 65 areas of the cat's cerebral cortex. For every pair of cortical areas, the connection strength is set to three if connections are strong or dense, two for the intermediate case, one if connections are weak or sparse, and zero if connections are absent or if no data had been reported. The

	C	ν	P	C	ν	P
	thyroid			heart		
RBF	6.4	9.4	**5.4**	21.4	19.1	22.4
POL	22.8	12.6	13.3	20.4	20.4	23.0
PLU	6.1	6.2	*5.7*	**16.3**	**16.3**	*17.4*
	breast cancer			banana		
RBF	33.6	31.6	32.4	*13.2*	36.7	**11.6**
POL	36.0	**25.7**	*27.1*	35.3	35.0	22.4
PLU	33.4	33.1	30.6	15.7	15.7	21.9
	german					
RBF	28.7	29.3	*27.8*			
POL	33.7	29.6	31.8			
PLU	28.8	28.5	**27.1**			

Table 1: Average percentage of misclassification for the UCI and the "banana" data sets. The table compares results obtained with the C-, ν-, and P-SVM for the Radial Basis Function (RBF), $\exp(-\frac{1}{2\ \sigma^2}\ \|\boldsymbol{x}^i\ -\ \boldsymbol{x}^j\|^2$, polynomial (POL), $\left(\left\langle \boldsymbol{x}^i, \boldsymbol{x}^j \right\rangle\ +\ \eta\right)^\delta$, and Plummer (PLU), $\frac{1}{(\|\boldsymbol{x}^i\ -\ \boldsymbol{x}^j\|\ +\ \rho)^\varsigma}$, kernels. Results were averaged over 100 experiments with separate training and test sets. For each data set numbers in bold and italic highlight the best and the second best result. The parameters C and ν for the SVM as well as the kernel parameters were determined using 5–fold cross validation on the training set then the SVM was trained on the training set and tested on the test set. The parameters are different for the individual experiments.

values are summarized by a connectivity matrix, whose diagonal was set to four for "self-connections". All areas are labeled according to whether they belong to the auditory ("A"), visual ("V"), somatosensory ("SS"), or frontolimbic ("FL") systems.

Fig. 8 shows a scree plot of the eigenvalues of the connectivity matrix. It is not positive definite because it contains negative eigenvalues, and cannot be directly used as a Gram matrix in standard SVM methods. Table 2 summarizes classification results which were obtained with the generalized SVM (G-SVM, Graepel et al., 1999; Mangasarian, 1998) and the P-SVM method. While the P-SVM operates on the indefinite Gram matrix, the G-SVM interprets the "column" vectors of the Gram matrix as feature vectors. The table shows the percentage of misclassification for the four two-class classification problems "one class against the rest". The P-SVM yields slightly better classification results compared to the G-SVM but with considerably fewer support vectors. Therefore, the number of measurements needed in order to be able to classify a new data point (the number of support vectors) is much lower on average: 28 (classification task "V"), 19 (classification task "A"), and 33 (classification task "SS"), and 35 (classification task "FL") compared to the 65 measurements always needed when using the G-SVM.

cat cortex					
	Reg.	V	A	SS	FL
Size	—	18	10	18	19
G-SVM	0.05	4.6	3.1	**3.1**	**1.5**
G-SVM	0.1	4.6	3.1	6.1	**1.5**
G-SVM	0.2	6.1	**1.5**	**3.1**	3.1
P-SVM	0.6	**3.1**	**1.5**	6.1	3.1
P-SVM	0.7	**3.1**	3.1	4.6	**1.5**
P-SVM	0.8	**3.1**	3.1	4.6	**1.5**

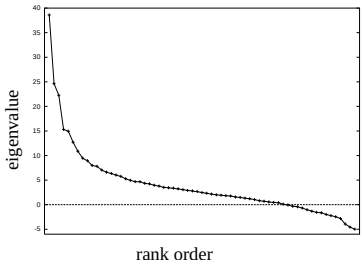

Table 2 (left): Percentage of misclassification for the "cat cortex" data set for classifiers obtained with the P-SVM and G-SVM methods. Column "Reg." lists the value of the regularization parameter (ν for G-SVM and C for P-SVM). Columns "V" to "FL" provide the results for the four classification problems "one class against the rest". The percentage of misclassification was computed using leave-one-out cross validation. The best classification results for each problem are shown in bold. Figure 8 (right): Scree plot of the eigenvalues of the "cat cortex" data set. The positive eigenvalues are more prominent than the negative one's so that projection into the subspace spanned by the positive eigenvalues or flipping the sign of the negative eigenvalues also leads to acceptable classification results (Graepel et al., 1999).

3.1.3 Protein Data Set

The "protein" data set (cf. Hofmann and Buhmann, 1997) was provided by M. Vingron and consists of 226 proteins from the globin families. Pairs of proteins are characterized by their evolutionary distance, which is defined as the probability of transforming one amino acid sequence into the other via point mutations. Class labels are provided, which denote membership in one of the four families: hemoglobin-α ("H-α"), hemoglobin-β ("H-β"), myoglobin ("M"), and heterogenous globins ("GH").

Table 3 summarizes the classification results, which were obtained with the G-SVM and the P-SVM methods (see Section 3.1.2). The table shows the percentage of misclassification for the four two-class classification problems "one class against the rest". Again, the P-SVM yields better classification results using on average 180 proteins (support vectors) compared to 203 protein used by the G-SVM (note that for 10–fold cross validation 203 is the average training size). Here small number of support vectors is highly desirable, because it reduces the computational costs of sequence alignments which are necessary for the classification of new examples.

3.1.4 World Wide Web Data Set

The "World Wide Web" data sets consist of 8,282 WWW-pages collected during the Web⟶Kb project at Carnegie Mellon University in January 1997 from

protein data					
	Reg.	H-α	H-β	M	GH
Size	—	72	72	39	30
G-SVM	0.05	1.3	4.0	0.5	0.5
G-SVM	0.1	1.8	4.5	0.5	0.9
G-SVM	0.2	2.2	8.9	0.5	0.9
P-SVM	300	**0.4**	3.5	**0.0**	**0.4**
P-SVM	400	**0.4**	**3.1**	**0.0**	0.9
P-SVM	500	**0.4**	3.5	**0.0**	1.3

Table 3: Percentage of misclassification for the "protein" data set. The entries are as in Table 2 except that the generalization error is now measured by 10–fold cross validation.

the web sites of the computer science departments of the four universities Cornell University ("Cornell"), Texas University ("Texas"), Washington University ("Washington"), and Wisconsin University ("Wisconsin"). The pages were manually classified into the categories "student", "faculty", "staff", "department", "course", "project", and "other".

Every pair (i, j) of pages is characterized by whether page i contains a hyperlink to page j and vice versa. The data is summarized using two binary matrices and a ternary matrix. The first matrix $\boldsymbol{K}$ ("out") contains a one for at least one outgoing link $(i \to j)$ and a zero if no outgoing link exists, the second matrix $\boldsymbol{K}^\top$ ("in") contains a one for at least one ingoing link $(j \to i)$ and a zero otherwise, and the third, ternary matrix $\frac{1}{2}\left(\boldsymbol{K} + \boldsymbol{K}^\top\right)$ ("sym") contains a zero, if no link exists, a value of 0.5, if only one unidirectional link exists, and a value of 1, if links exists in both directions.

For the following experiments, we restricted the data set to pages from first six classes which had more than one in- or outgoing link. The data set thus consists of the four subsets "Cornell" (350 pages), "Texas" (286 pages), "Wisconsin" (300 pages), and "Washington" (433 pages).

Table 4 summarizes the classification results for the G- and P-SVM methods. The parameter C for both SVMs was optimized for each cross validation trial using another 4–fold cross validation on the training set. Again, the P-SVM provides better results with fewer support vectors. Interestingly, classification results are better for the asymmetric matrices "in" and "out" than for the symmetric matrix "sym". The reason for the better performance of asymmetric matrices is that in some cases very indicative pages (hubs) exist which are connected to one particular class of pages by either in- or outgoing links. At Cornell university, for example, the project pages have indicative outgoing links and the Texas university contains web pages which are indicative for the student pages by linking only them. The symmetric case blurs the contribution of the indicative pages because ingoing and outgoing links can no longer be distinguished which leads to poorer performance. Because the P-SVM yields fewer support

	Course	Faculty	Project	Student
Cornell University				
Size	57	60	52	143
G-SVM (sym)	24	43	46	44
P-SVM (sym)	17	25	16	32
P-SVM (out)	**14**	23	**13**	28
P-SVM (in)	15	**21**	28	**27**
Texas University				
Size	52	35	29	129
G-SVM (sym)	28	35	34	59
P-SVM (sym)	16	14	12	26
P-SVM (out)	**8**	**10**	**10**	21
P-SVM (in)	12	**10**	9	**13**
Wisconsin University				
Size	77	36	22	117
G-SVM (sym)	37	37	19	54
P-SVM (sym)	19	15	10	34
P-SVM (out)	**12**	11	8	24
P-SVM (in)	13	**9**	**6**	**13**
Washington University				
Size	169	44	39	151
G-SVM (sym)	19	27	22	37
P-SVM (sym)	17	13	9	20
P-SVM (out)	**11**	13	**7**	17
P-SVM (in)	12	**9**	**7**	**14**

Table 4: Percentage of misclassification for the World Wide Web data sets for classifiers obtained with the P-SVM and G-SVM methods. The percentage of misclassification was measured using 10–fold cross-validation. The best results for each data set and classification task are indicated in bold. Values for the hyperparameter C varied because we chose them for each cross validation trial through another 4–fold cross validation on the training set. For details see text.

vectors, online classification is faster than for the G-SVM: Fewer "row" support vector pages have to be analyzed in order to classify a new page. Another advantage is that if web pages cease to exist, the P-SVM is more likely not to be affected because only "support" web pages matter which are assumed to exist longer.

Table 5 provides a more detailed analysis of the classification results for the problem "student pages vs. the rest". The false positive rate for the matrix "out" is higher than the matrix "in". This means that the most indicative pages, which are referred by "student" pages, are not as discriminative as pages indexing student pages.

	pages	student pages	P-SVM "in"	P-SVM "out"	P-SVM "sym"	G-SVM "sym"
Cornell	350	143	27 +21/-32	28 +22/-33	32 +42/-25	44 +51/-40
Texas	286	129	13 +10/-15	21 +35/-9	26 +33/-20	59 +48/-68
Wisconsin	300	117	13 +14/-13	24 +32/-19	34 +42/-29	54 +51/-56
Washington	433	151	14 +18/-12	17 +36/-7	20 +30/-15	37 +36/-38

Table 5: Classification results for the problem "student pages vs. the rest" for the "world wide web" data set. The percentage of misclassifications is analyzed with respect to the false positive rate ("+") and the false negative rate ("-"). Unsigned numbers in the rightmost four columns denote the total percentage of errors.

3.2 Application to Regression Problems

In this section we report results for the data sets "robot arm" (2 features), "boston housing" (13 features), "computer activity" (21 features), and "abalone" (10 features) data sets from the UCI benchmark repository. The data preprocessing is described in (Chu et al., 2004), and the data sets are available as training / test set pairs at `http://guppy.mpe.nus.edu.sg/~chuwei/data`. The size of the data sets were (training set / test set): "robot arm": 200 / 200, 1 set; "boston housing": 481 / 25, 100 sets; "computer activity": 1000 / 6192, 10 sets; "abalone": 3000 / 1177, 10 sets.

Pairwise data sets were generated by constructing the Gram matrices for Radial Basis Function kernels of different widths σ, and the Gram matrices were used as input for the classification methods: C-support vector regression (SVR) (Schölkopf and Smola, 2002), Bayesian support vector regression (BSVR) (Chu et al., 2004), and the P-SVM. Hyperparameters (C and σ) were optimized using n-fold cross-validation ($n = 50$ for "robot arm", $n = 20$ for "boston housing"; $n = 4$ for "computer activity" and $n = 4$ for "abalone"). Parameters were first optimized on a coarse 4×4 grid and later refined on a 7×7 fine grid around the values for C and σ selected in the first step (65 tests per parameter selection).

Table 6 shows regression results. It shows the mean squared error and its standard deviation for the various combinations of training and test set. Except for the "robot arm" data set, the P-SVM method provides the best regression results, and even in the robot arm case the P-SVM result is only insignificantly worse than the result obtained with the other methods. Note, that the P-SVM results also have lower variance of the prediction error. These results show that the P-SVM is competitive to and in many cases better than state-of-the-art regression methods.

	SVR	BSVR	P-SVM
robot arm (10^{-3})	**5.84**	5.89	5.88
boston housing	10.27±7.21	12.34±9.20	**9.42**±4.96
computer activity	13.80±0.93	17.59±0.98	**10.28**±0.44
abalone	0.441±0.021	0.438±0.024	**0.424**±0.017

Table 6: Regression results for the UCI data sets. The table shows the mean squared error ± standard deviation. Best results for each data set are shown in bold. Note, that for robot arm only one data set was available and, therefore, no standard deviation is given. For details see text.

3.3 Application to Feature Selection Problems

In this section we apply the P-SVM to feature selection problems of various kinds, using the "correlation threshold" regularization scheme (Section 2.4.3). This section consists of three parts. In the first part, we reanalyze the "protein" and "world wide web" data sets of sections 3.1.4 and 3.1.3. However both regularization schemes are now used simultaneously. In the second and third part we specifically apply the P-SVM to data sets from DNA microarray experiments. These kinds of data sets provide a challenge to classification, regression, and feature selection techniques, because they are characterized by a small number of samples a high level of measurement noise, and an extremely high number of features (genes), from which only a small number of features are actually indicative of the samples attributes. In Section 3.3.2 we construct a DNA microarray toy data set based on ideas from (Weston et al., 2000) in order to asses the performance of the P-SVM in comparison to several statistical and kernel-based methods. Artificial data allows us to interpret the success of the different methods by comparing the selected features to the features which were indeed indicative of the sample class. In Section 3.3.3, finally, three real-world microarray data sets are considered, and the P-SVM feature selection method is judged by how well a standard classification technique performs using the set of selected features as the object's description. Details with respect to the P-SVM as a feature selection method and more information concerning data sets and numerical experiments can be found in (Hochreiter and Obermayer, 2004a).

3.3.1 Protein and World Wide Web Data Sets

In this section we again apply the P-SVM to the "protein" and "world wide web" data sets of sections 3.1.4 and 3.1.3. Using both regularization schemes simultaneously leads to a trade-off between a small number of features (a small number of measurements) and a good classification result. Reducing the number of features is beneficial if measurements are costly and a small increase in prediction error can be tolerated.

Table 7 shows the results for the "protein" data sets for various values of the regularization parameter ϵ. C was set to 100, because it gave good results

protein data				
ϵ	H-α	H-β	M	GH
0.2	1.3	4.9	0.9	1.3
	(203)	(203)	(203)	(203)
1	2.6	5.3	1.3	4.4
	(41)	(110)	(28)	(41)
10	3.5	8.8	1.8	13.3
	(10)	(26)	(5)	(7)
20	3.5	8.4	4.0	13.3
	(5)	(12)	(4)	(5)

Table 7: Percentage of misclassification and the number of support features (in brackets) for the "protein" data set for the P-SVM method. The maximum number of features is 226. The value for ϵ is provided in the first column (C was 100). The four columns to the right show the results for the four classification problems "one class against the rest" using 10–fold cross-validation.

for a wide range of ϵ values. We chose a minimal $\epsilon = 0.2$ because it resulted in a classifier, where all complex features were support vectors (for 10–fold cross validation 203 is the training set size). Note, that C was smaller than in the experiments in Section 3.1.3 because large ϵ values pushed the dual variables $\boldsymbol{\alpha}$ towards zero and, therefore, large C values have no influence. The table shows that classification performance drops if less features are considered, but that 5 % of the features suffice to obtain a performance which lead only to about 5 % misclassification compared to about 2 % at the optimum. Since every feature value has to be determined via a sequence alignment, this saving in computation time might is essential for large data bases like the Swiss-Prot data base (130,000 proteins), where supplying all pairwise relations is currently impossible.

Table 8 shows the corresponding results (10–fold cross validation) for the P-SVM applied to the "world wide web" data set "Cornell" and for the classification problem "student pages vs. the rest". Only ingoing links (matrix $\boldsymbol{K}^\top$ of Section 3.1.4) were used. P-SVM hyperparameters C were optimized using 3–fold cross validation on the corresponding training sets for each of the 10–fold cross validation runs. By increasing the regularization parameter ϵ the number of web pages which have to be considered in order to classify a new page (the number of support vectors) decreases from 135 to 8. At the same time the percentage of pages which can no longer be classified because they receive no ingoing link from one of the "support vector page" increases. The percentage of misclassification, however, is reduces from 14 % for $\epsilon = 0.1$ to 0.6 % for $\epsilon = 2.0$. With only 8 pages providing ingoing links more than 50 % of the pages could be classified with only 0.6 % misclassification rate.

"Cornell" data set, student pages			
ϵ	% classified	% incorrect	# (%) SVs
0.1	84	14	135 (38.6)
0.2	81	12	115 (32.8)
0.3	79	9.7	99 (28.3)
0.4	75	6.9	72 (20.6)
0.5	73	5.5	58 (16.6)
0.6	71	4.8	48 (13.7)
0.7	66	3.9	38 (10.9)
0.8	65	3.1	34 (9.7)
0.9	64	2.7	32 (9.1)
1.0	61	1.4	27 (7.7)
1.1	59	1.0	21 (6.0)
1.4	56	1.0	12 (3.4)
1.6	55	1.0	10 (2.8)
2.0	51	0.6	8 (2.3)

Table 8: Feature selection and classification results of 10–fold cross validation for the P-SVM method for "world wide web" data set "Cornell" and the classification problem "student pages against the rest". The first column shows the chosen ϵ for the P-SVM (C was optimized through a 3–fold cross validation on the corresponding training set). Columns three to five show the percentage of classified pages, the percentage of misclassifications and the number (percentage) of support vectors.

3.3.2 Weston Data Set

In this section we consider one toy data set similar to, but more difficult than the data set used in the feature selection study of Weston et al. (2000). The data set is generated in order to provide a model for the data recorded in typical DNA microarray experiments.

Feature selection is performed in the context of a binary classification task, where "column" objects fall into one of two classes. Every object is described by its relationship with a large number of "row" objects or features (the "genes"). For the following numerical experiments we choose 600 "column" objects, 300 objects from each class. 100 "column" objects were used for feature and model selection and the remaining 500 "column" objects are the test set. Every "column" object was characterized by its relationship to 2000 "row" objects or "complex features". Four out of the first 20 features were indicators of the class membership, all remaining 1980 features were not correlated with the class of the "column" objects. The first 20 features were grouped into the five modes 1–4 ($l = 0$), 5–8 ($l = 4$), 9–12 ($l = 8$), 13–16 ($l = 12$), 17–20 ($l = 16$). For every "column object" $\boldsymbol{x}^i$ (sample) a label $y_i \in \{+1, -1\}$ was chosen with probability 0.5 for $+1$ and probability 0.5 for -1, then one mode $l \in \{0, 4, 8, 12, 16\}$ was chosen

no. of features	5	10	15	20	30
Fisher	0.31	0.28	0.26	0.25	0.26
RFE	0.33	0.32	0.32	0.31	0.32
R2W2	0.29	0.28	0.28	0.27	0.27
P-SVM	0.28	0.23	0.24	0.24	0.26

Table 9: Classification performance for the "Weston" data set described in the text. The values are the fractions of misclassification averaged over 10 runs on different test sets for classifiers trained on the selected features. The table shows the results using the top ranked 5, 10, 15, 20, and 30 features for the methods: Fisher statistics (Kendall and Stuart, 1977), Recursive Feature Elimination (RFE), R2W2, and the P-SVM.

with probability 0.2 for each value of l. Then the values of the four associated features $x^i_{l+\tau}$, $1 \leq \tau \leq 4$, were chosen according to $x^i_{l+\tau} \sim y_i \cdot \mathrm{N}(2, 0.5\,\tau)$. The remaining features from 1 to 20 (that is excluding the features $x^i_{l+\tau}$, $1 \leq \tau \leq 4$) were chosen according to $x^i_j \sim \mathrm{N}(0,1)$, $1 \leq j \leq 20$, $j \neq l+\tau$. Finally, the remaining 1980 features which were never indicative of class membership were chosen according to $x^i_j \sim \mathrm{N}(0, 20)$, $21 \leq j \leq 2000$.

This data set has the typical structure of DNA microarray data: many features (2000), few indicative features (20), and few training examples (100). The data set was then analyzed by a two-stage procedure. In the first stage, four feature selection methods were applied in order to separate potential indicative features from the irrelevant ones. These methods were the P-SVM (Section 2.4.3), the Fisher statistics (Kendall and Stuart, 1977), Recursive Feature Elimination (RFE) method of Guyon et al. (2002), and the R2W2 method (Weston et al., 2000). All methods rank the importance of features, where ranking is based on the support vector weights for the P-SVM method, the class discriminant value for Fisher statistics, on multiple runs for RFE (Guyon et al., 2002), and on the feature scaling factors for R2W2 (Weston et al., 2000). In the second step, we treated the columns of the data matrix as feature vectors and used these feature vectors — which included only the top ranked 5, 10, 15, 20, and 30 features selected in step 1 — as input into a standard C-SVM. The hyperparameter C was selected through 5-fold cross-validation on the training set from the set $\{0.01, 0.1, 1, 10, 100\}$ for all methods ($C = 0.1$ was chosen in most cases). The table shows that the P-SVM method performed best.

The success of feature selection depends on how many irrelevant features are wrongly selected because of noise and whether all modes which influence classification performance are represented sufficiently well. It is instructive to compare the results of Table 9 with the prediction quality of a classifier trained using the 20 relevant features (perfect selection), which leads to a fractional error of 0.10, and using all 2000 features (no selection), which leads to a fractional error of 0.38. Feature selection improves the classification result but does not quite reach the performance of the "perfect selection" case because not all

P-SVM:	**7**	837	**2**	**18**	1248	**5**	**6**	**12**
	20	**14**	1562	980	664	1110	**11**	1404
	1822	668	525	**9**	80	1205	997	1228
	1331	289	1605	621	1277	1987		
R2W2:	837	**2**	980	**7**	**20**	**11**	1277	**6**
	45	**5**	**18**	1822	**12**	621	398	664
	289	**14**	1110	587	1605	1833	1331	1248
	1752	525	1060	1443	820	997		
Fisher:	980	**7**	**5**	837	**6**	**18**	1562	**12**
	2	837	**20**	1248	**8**	1404	**14**	1110
	11	1228	80	664	1987	1275	1331	668
	263	640	621	1954	1774	1605		
RFE:	837	**7**	1987	1277	**2**	753	**20**	1110
	1774	997	219	1636	**12**	398	**6**	1472
	536	820	**18**	314	974	525	**14**	877
	621	1516	540	654	1331	664		

Table 10: Numbers of the top 30 selected features for a typical single trial, listed according to their rank. Indicative features, i.e. features from the set 1 to 20, are printed in boldface.

relevant features were selected. R2W2 with the weighting coefficients instead of selecting features has an error of 0.26, that means R2W2 in the non-selection mode is better than in the selection mode.

Table 10 shows the numbers of the top 30 selected features for a typical single trial, listed according to their rank. P-SVM found 11, R2W2 9, Fisher statistics 10, and RFE 7 relevant features (numbers printed in boldface). All other features are spurious and were selected because the high level of noise and the small number of samples led to spurious correlations between values of these features and the residual error. All five modes were detected by P-SVM, R2W2, Fisher statistics, and RFE using the 10, 18, 15, and 23 most highly ranked features. P-SVM detected indicator features corresponding to all five modes using the smallest features set from all the methods tested. This explains the better performance of classifiers based on the P-SVM feature set.

3.3.3 Microarray Data Sets

In this subsection we apply the P-SVM to real DNA microarray data. The data was taken from Pomeroy et al. (2002), Shipp et al. (2002), and van't Veer et al. (2002). The P-SVM results are taken from (Hochreiter and Obermayer, 2004a) where the details concerning the data sets and the gene selection procedure based on the P-SVM can be found. The data sets were:

1. *Brain tumor data set (Pomeroy et al., 2002).* The data consists of 60 tissue samples of human brain tumors which were characterized by the expres-

sion values of 7129 genes[4]. Samples were labeled according to whether the particular tumor responded favorably or unfavorably to a particular treatment.

2. *Lymphoma data set (Shipp et al., 2002).* The data set consists of 58 samples from human lymphoma tumors characterized by the expression values of 7129 genes. The samples were labeled according to the treatment outcome.

3. *Breast cancer data set (van't Veer et al., 2002).* The data set consists of 78 tissue samples of human breast cancer characterized by the expression values of 24481 genes. The tissue samples were labeled according to the treatment outcome.

We used the gene selection protocol from (Hochreiter and Obermayer, 2004a) which is based on multiple runs of the P-SVM to obtain stable results. Details of the gene selection protocol can be found there. After feature selection a *linear* ν-SVM (Schölkopf and Smola, 2002) with offset $b = 0$ was used for selecting a classifier in order to predict the outcome of the medical treatment. The hyperparameter ν from the set $\{0.2, 0.3, 0.4, 0.5\}$ was optimized with cross validation on the training set according to the gene selection protocol. The methods chosen for the benchmark were:

	selection method	classification method
(1)	expression value of the TrkC gene	one gene classification
(2)	SPLASH (Califano et al., 1999)	likelihood ratio classifier (LRC)
(3)	signal-to-noise-statistics (STN)	K-nearest neighbor (KNN)
(4)	signal-to-noise-statistics (STN)	weighted voting (voting)
(5)	Fisher statistics (Fisher)	weighted voting (voting)
(6)	R2W2	R2W2
(7)	P-SVM	ν-SVM

	references
(1)	Pomeroy et al., 2002
(2)	Pomeroy et al., 2002
(3)	Pomeroy et al., 2002; Shipp et al., 2002
(4)	Pomeroy et al., 2002; Shipp et al., 2002
(5)	van't Veer et al., 2002
(6)	Pomeroy et al., 2002; Shipp et al., 2002
(7)	Hochreiter and Obermayer, 2004a

The results which are taken from the corresponding literature are summarized in Table 11. The P-SVM method identified a smaller number of genes except for the "lymphoma" data set. The final classification results show that the P-SVM

[4]Actually, the microarray chip contains 7129 probes which do not allow a one-to-one mapping to genes. Probes may indicate the expression of more than one gene, serve control purposes, or one gene is indicated by more than one probe. For simplicity we associate one probe with one gene and its expression value.

Brain Tumor

Feature Selection / Classification	# F	# E
TrkC (one gene)	1	33
SPLASH / LRC	–	25
R2W2	*	25
STN / voting	–	23
STN / KNN	8	22
TrkC & SVM & KNN	–	20
P-SVM / ν-SVM	45	**7**

Lymphoma

Feature Selection / Classification	# F	# E
STN / KNN	8	28
STN / voting	13	24
R2W2	*	22
P-SVM / ν-SVM	18	**21**

Breast Cancer

Feature Selection / Classification	# F	# E	ROC area
Fisher / voting	70	26	0.88
P-SVM / ν-SVM	30	**15**	**0.77**

Table 11: Feature selection and classification results for the "brain tumor", "lymphoma", and "breast cancer" data sets. The table shows the leave-one-out error E (% misclassifications) and the number F of features. For breast cancer "E" gives the minimal leave-one-out error over different threshold values (in van't Veer et al., 2002, only this error is given for comparison). Therefore, the area under a receiver operating curve (ROC) is provided which was calculated by varying the threshold b of the classifier (Hochreiter and Obermayer, 2004a). For R2W2 "*" means that there is no "number of features" because R2W2 scales features rather than selecting them. The hyperparameter ν for P-SVM / ν-SVM was chosen through cross validation on the training set according to the gene selection protocol (Hochreiter and Obermayer, 2004a). For further explanation see text.

method clearly outperforms standard methods — for the "brain tumor" data set the number of misclassifications is down by a factor of 3.

4 Summary

In this contribution we have described the Potential Support Vector Machine (P-SVM) as new method for classification, regression, and feature selection. The P-SVM selects models using the principle of structural risk minimization. In contrast to standard SVM approaches, however, the P-SVM is based on a new objective function and a new set of constraints which lead to an expansion of the classification or regression function in terms of "support features". The

combination of the new objective with the new constraints results in a quadratic problem which is always well defined, suited for data in matrix form, and neither requires square nor positive definite Gram matrices. Therefore, the method can also be used with matrices which are measured rather than being constructed using a vectorial representation and a kernel function. In feature selection mode the P-SVM allows to select and rank the features through the support vector weights of its sparse set of support vectors. The sparseness constraint avoids the construction of sets for features, which are redundant. In a classification or regression setting this is a clear advantage over statistical methods where redundant features are often kept as long as they provide information about the objects' attributes. Because the dual formulation of the optimization problem can be solved by a fast sequential minimal optimization technique, the new P-SVM can be applied to data sets with many features. Compared to state-of-the-art classification, regression and feature selection methods, the P-SVM provided the best results.

Finally, we have suggested a new interpretation of data in matrix form. Objects in real world are no longer described by vectorial representations. Structures like dot products or norms are induced directly through measurements of object pairs, i.e. through relations between objects. This opens up a new field of research where relations between real world objects determine mathematical structures.

Acknowledgments

We thank Merlyn Albery-Speyer, Christoph Büscher, Cyril Minoux, Raman Sanyal, and Sambu Seo for their help with the numerical simulations. This work was funded by the DFG (SFB 618) and the Anna-Geissler-Stiftung.

A Proof of the Simplified Expression for b

In this appendix we prove the simplified expression for b, eq. (26):

$$b \;=\; \frac{1}{L} \sum_{i=1}^{L} y_i \;. \tag{47}$$

Proof.
If for the optimal $\boldsymbol{w}$ follows that $\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1} \;=\; 0$ then inserting this equality into eq. (17) gives eq. (48). To prove $\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1} \;=\; 0$, we need two properties of the pseudo inverse (Moore-Penrose inverse) $\boldsymbol{X}^*$ of the matrix $\boldsymbol{X}$ (Lütkepohl, 1996):

$$\boldsymbol{X} \;=\; \boldsymbol{X}\,\boldsymbol{X}^\top \left(\boldsymbol{X}^\top\right)^* \quad \text{and} \tag{48}$$

$$\boldsymbol{X}^\top \;=\; \boldsymbol{X}^*\,\boldsymbol{X}\,\boldsymbol{X}^\top \;. \tag{49}$$

We assume that $\boldsymbol{w}$ is the solution of problem eqs. (25) and that $\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1} \;\neq\; 0$. A value ζ can be chosen such that

$$\begin{aligned} \zeta \;&>\; \mathbf{1}^\top \boldsymbol{X}^\top \left(\boldsymbol{X}^\top\right)^* \mathbf{1} \;=\; \mathbf{1}^\top \boldsymbol{X}^*\,\boldsymbol{X}\,\boldsymbol{X}^\top \left(\boldsymbol{X}^\top\right)^* \mathbf{1} \\ &=\; \left((\boldsymbol{X}^*\boldsymbol{X})^\top \mathbf{1}\right)^\top \left((\boldsymbol{X}^*\boldsymbol{X})^\top \mathbf{1}\right) \;\geq\; 0 \;, \end{aligned} \tag{50}$$

where we applied eq. (50) and $\left(\boldsymbol{X}^\top\right)^* \;=\; \left(\boldsymbol{X}^*\right)^\top$. From $\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1} \;\neq\; 0$ follows that $\boldsymbol{X}\,\mathbf{1} \;\neq\; \mathbf{0}$ and, because of $\boldsymbol{X}\,\mathbf{1} \;=\; \boldsymbol{X}\,\boldsymbol{X}^\top \left(\boldsymbol{X}^\top\right)^* \mathbf{1}$ (cf. eq. (49)), also $\left(\boldsymbol{X}^\top\right)^* \mathbf{1} \;\neq\; \mathbf{0}$. Now, we define a vector $\boldsymbol{u} \;\neq\; \boldsymbol{w}$ as

$$\boldsymbol{u} \;:=\; \boldsymbol{w} \;-\; \frac{\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}}{\zeta} \left(\boldsymbol{X}^\top\right)^* \mathbf{1} \tag{51}$$

and obtain

$$\begin{aligned} &\boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{u} \;-\; \boldsymbol{y}\right) \\ &=\; \boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} \;-\; \boldsymbol{y}\right) \;-\; \frac{\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}}{\zeta}\, \boldsymbol{K}^\top \boldsymbol{X}^\top \left(\boldsymbol{X}^\top\right)^* \mathbf{1} \\ &=\; 0 \;-\; \frac{\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}}{\zeta}\, \boldsymbol{Z}^\top \boldsymbol{X}\;\; \boldsymbol{X}^\top \left(\boldsymbol{X}^\top\right)^* \mathbf{1} \;=\; -\; \frac{\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}}{\zeta}\, \boldsymbol{Z}^\top \boldsymbol{X}\,\mathbf{1} \\ &=\; -\; \frac{\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}}{\zeta}\, \boldsymbol{K}^\top \mathbf{1} \;=\; 0 \;, \end{aligned}$$

where we used the fact that $\boldsymbol{w}$ fulfills the constraints, $\boldsymbol{K}^\top \mathbf{1} \;=\; \mathbf{0}$ (cf. eqs. (23)),

and eq. (49). We also obtain

$$
\begin{aligned}
\|\boldsymbol{X}^\top \boldsymbol{u}\|^2 \;&=\; \|\boldsymbol{X}^\top \boldsymbol{w} \;-\; \frac{\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}}{\zeta}\, \boldsymbol{X}^\top \left(\boldsymbol{X}^\top\right)^* \mathbf{1}\|^2 \qquad (52)\\
&=\; \|\boldsymbol{X}^\top \boldsymbol{w}\|^2 - \frac{\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}}{\zeta}\, \boldsymbol{w}^\top \boldsymbol{X}\, \boldsymbol{X}^\top \left(\boldsymbol{X}^\top\right)^* \mathbf{1} \;-\\
&\frac{\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}}{\zeta}\, \mathbf{1}^\top \boldsymbol{X}^*\, \boldsymbol{X}\, \boldsymbol{X}^T \boldsymbol{w}\\
&+\; \left(\frac{\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}}{\zeta}\right)^2 \mathbf{1}^\top \boldsymbol{X}^*\, \boldsymbol{X}\, \boldsymbol{X}^T \left(\boldsymbol{X}^\top\right)^* \mathbf{1}\\
&=\; \|\boldsymbol{X}^\top \boldsymbol{w}\|^2 \;-\; 2\,\frac{\left(\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}\right)^2}{\zeta} \;+\; \frac{\left(\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}\right)^2}{\zeta}\, \frac{\mathbf{1}^\top \boldsymbol{X}^\top \left(\boldsymbol{X}^\top\right)^* \mathbf{1}}{\zeta}\\
&\leq\; \|\boldsymbol{X}^\top \boldsymbol{w}\|^2 \;-\; \frac{\left(\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1}\right)^2}{\zeta} \;<\; \|\boldsymbol{X}^\top \boldsymbol{w}\|^2\,,
\end{aligned}
$$

where we applied both eq. (49) and eq. (50) as well as eq. (51). We constructed a vector $\boldsymbol{u}$ which fulfills the constraints but leads to a smaller value of the objective function than $\boldsymbol{w}$, in contradiction to the assumption that $\boldsymbol{w}$ was the solution of the optimization problem eqs. (25). Therefore, we showed that $\boldsymbol{w}^\top \boldsymbol{X}\,\mathbf{1} = 0$ holds for the solution $\boldsymbol{w}$ of problem eqs. (25).
■

B The Sequential Minimal Optimization (SMO) Technique for the P-SVM Method

The Sequential Minimal Optimization (SMO) algorithm was introduced by (Platt, 1999) as a fast method for solving the dual optimization problem of support vector machines. Here we describe a modified version of the SMO which can be applied to the general P-SVM optimization problem given by the primal

$$
\begin{aligned}
\min_{\boldsymbol{w},\boldsymbol{\xi}^+,\boldsymbol{\xi}^-} \quad & \frac{1}{2}\,\|\boldsymbol{X}^\top \boldsymbol{w}\|^2 \;+\; C\mathbf{1}^\top \left(\boldsymbol{\xi}^+ \;+\; \boldsymbol{\xi}^-\right) \qquad (53)\\
\text{s.t.} \quad & \boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} \;-\; \boldsymbol{y}\right) \;+\; \boldsymbol{\xi}^+ \;+\; \epsilon \;\geq\; \mathbf{0}\\
& \boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} \;-\; \boldsymbol{y}\right) \;-\; \boldsymbol{\xi}^- \;-\; \epsilon \;\leq\; \mathbf{0}\\
& \mathbf{0} \;\leq\; \boldsymbol{\xi}^+, \boldsymbol{\xi}^-
\end{aligned}
$$

and the dual

$$
\begin{aligned}
\min_{\boldsymbol{\alpha}^+,\boldsymbol{\alpha}^-} \quad & \frac{1}{2}\left(\boldsymbol{\alpha}^+ \;-\; \boldsymbol{\alpha}^-\right)^\top \boldsymbol{K}^\top \boldsymbol{K}\, \left(\boldsymbol{\alpha}^+ \;-\; \boldsymbol{\alpha}^-\right) \qquad (54)\\
& \;-\; \boldsymbol{y}^\top \boldsymbol{K}\, \left(\boldsymbol{\alpha}^+ \;-\; \boldsymbol{\alpha}^-\right) \;+\; \epsilon\, \mathbf{1}^\top \left(\boldsymbol{\alpha}^+ \;+\; \boldsymbol{\alpha}^-\right)\\
\text{s.t.} \quad & \mathbf{0} \;\leq\; \boldsymbol{\alpha}^+, \boldsymbol{\alpha}^- \;\leq\; C\mathbf{1}\;\;.
\end{aligned}
$$

We will use the following notation:

$$
\begin{aligned}
Q_{i,j} \;&=\; \sum_{l=1}^{L} K_{l,i}\; K_{l,j}\;, \quad \boldsymbol{Q} \;=\; \boldsymbol{K}^\top \boldsymbol{K}\;, \\
l_j \;&=\; \sum_{l=1}^{L} K_{l,j}\; y_l\;, \quad \boldsymbol{l} \;=\; \boldsymbol{K}^\top \boldsymbol{y}\;, \\
F_j \;&:=\; [\boldsymbol{Q}\boldsymbol{\alpha}]_j \;-\; l_j\;, \\
F_j^+ \;&:=\; F_j \;+\; \epsilon, \\
F_j^- \;&:=\; -F_j \;+\; \epsilon,
\end{aligned}
\tag{55}
$$

where F_j is the error of the jth constraint.

The Karush-Kuhn-Tucker (KKT) conditions state that the product between dual variables and the primal constraints is zero for the optimal solution. Using

$$
\begin{aligned}
\boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} \;-\; \boldsymbol{y}\right) \;+\; \boldsymbol{\xi}^+ \;+\; \epsilon\, \mathbf{1} \;&=\; \boldsymbol{F}^+ \;+\; \boldsymbol{\xi}^+ \qquad \text{and} \\
\boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} \;-\; \boldsymbol{y}\right) \;-\; \boldsymbol{\xi}^- \;-\; \epsilon\, \mathbf{1} \;&=\; -\left(\boldsymbol{F}^- \;+\; \boldsymbol{\xi}^-\right)\;,
\end{aligned}
\tag{56}
$$

we obtain for the KKT conditions

$$
\begin{aligned}
&\alpha_j^+ \left(F_j^+ \;+\; \xi_j^+\right) \;=\; 0\;, \\
&-\,\alpha_j^- \left(F_j^- \;+\; \xi_j^-\right) \;=\; 0\;, \\
&\mu_j^+\, \xi_j^+ \;=\; 0\;, \text{ and} \\
&\mu_j^-\, \xi_j^- \;=\; 0\;.
\end{aligned}
\tag{57}
$$

The $\boldsymbol{\alpha}^+$ and $\boldsymbol{\alpha}^-$ are the Lagrange multipliers for the residual error constraints (the mixed moments), and $\boldsymbol{\mu}^+$ and $\boldsymbol{\mu}^-$ are the Lagrange multipliers for the slack variable constraints ($\boldsymbol{\xi}^+ \;\geq\; \mathbf{0}$ and $\boldsymbol{\xi}^- \;\geq\; \mathbf{0}$). The derivative of the Lagrangian with respect to $\boldsymbol{\xi}^+$ and $\boldsymbol{\xi}^-$ is zero for the solutions of eqs. (55) and (54), i.e.:

$$
\begin{aligned}
C\mathbf{1} \;-\; \boldsymbol{\alpha}^+ \;-\; \boldsymbol{\mu}^+ \;&=\; \mathbf{0} \quad \text{and} \\
C\mathbf{1} \;-\; \boldsymbol{\alpha}^- \;-\; \boldsymbol{\mu}^- \;&=\; \mathbf{0}\;.
\end{aligned}
\tag{58}
$$

Note that for $0 \;<\; \alpha_j^+$, it follows from the KKT conditions that

$$
\begin{aligned}
&\left[\boldsymbol{K}^\top \left(\boldsymbol{X}^\top \boldsymbol{w} \;-\; \boldsymbol{y}\right)\right]_j \;+\; \xi_j^+ \;+\; \epsilon \;=\; [\boldsymbol{Q}\boldsymbol{\alpha}]_j \;-\; l_j \;+\; \xi_j^+ \;+\; \epsilon \\
&=\; F_j^+ \;+\; \xi_j^+ \;=\; 0\;.
\end{aligned}
\tag{59}
$$

Because $F_j^+ \;+\; F_j^- \;=\; 2\epsilon$ we have $F_j^- \;=\; 2\epsilon \;+\; \xi_j^+$ at for the solution. That implies $F_j^- \;>\; 0$, and, therefore, $\alpha_j^- \;=\; 0$. Analogously, we deduce from $0 \;<\; \alpha_j^-$ that $\alpha_j^+ \;=\; 0$. Hence, for the solutions of eqs. (55) and (54) $\alpha_j^+ \cdot \alpha_j^- \;=\; 0$ is always fulfilled.

In the following, we first describe the SMO optimization (sections B.1 and B.2) and then the SMO variable selection step (Section B.3). In Section B.1 we treat the case that only the slack variables are used for regularization ($\epsilon \;=\; 0$)

whereas Section B.2 considers the case when correlation threshold regularization is used in addition. Section B.3 finally addresses how pairs of variables are chosen for every SMO iteration.

B.1 Optimization Step for Regularization with Slack Variables

For the regularization scheme based on slack variables (classification and regression), the dual optimization problem is given by

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha}} & \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{Q}\ \boldsymbol{\alpha}\ -\ \boldsymbol{l}^\top \boldsymbol{\alpha} \\ \text{s.t.} & -C\ \leq \alpha_j\ \leq\ C\ . \end{array} \tag{60}$$

We denote the objective function by O:

$$O\ :=\ \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{Q}\ \boldsymbol{\alpha}\ -\ \boldsymbol{l}^\top \boldsymbol{\alpha}\ . \tag{61}$$

Standard SMO proceeds iteratively by first choosing two variables α_1 and α_2 and then optimizing the Lagrangian for these two variables under the constraints. Since the usual equality constraint is missing in eqs. (61), it would suffice to optimize eqs. (61) with respect to one variable only in every iteration, i.e. it would suffice to optimize

$$\begin{aligned} O\left(\alpha_1\right)\ &=\ \frac{1}{2}\alpha_1^2\ Q_{11}\ +\ \alpha_1 \sum_{j\neq 1} \alpha_j\ Q_{1j}\ -\ \alpha_1\ l_1\ +\ c \\ &=\ -\ \frac{1}{2}\alpha_1^2\ Q_{11}\ +\ \alpha_1\ F_1 \end{aligned} \tag{62}$$

under constraint $-C\ \leq \alpha_1\ \leq\ C$ (c is a constant independent of α_1). The derivative of the objective with respect to $\alpha_j\ =\ \alpha_1$ must be zero

$$\frac{\partial O\left(\alpha_j\right)}{\partial \alpha_j}\ =\ F_j\ =\ 0\ , \tag{63}$$

where we used $\frac{\partial F_j}{\partial \alpha_j}\ =\ Q_{jj}$. Optimizing eqs. (61) for one variable at a time has the advantage that there is no need for an additional heuristics. Unfortunately, it turns out, that an SMO method based on eq. (63) is slow. This is particularly serious if the two rows $Q_{j\cdot}$ and $Q_{i\cdot}$ are similar but l_j and l_i differ. For $\xi_j\ =\ 0$ and $\xi_i\ =\ 0$, SMO attempts to fulfill the KKT conditions by setting F_j to zero, which changes an already zero F_i and oscillations between zeroing F_j and F_i may arise. Indeed we have observed strong oscillations of this kind in numerical simulations of ill-conditioned problems.

In order to avoid abovementioned problem, we suggest to optimize the objective function with respect to two variables α_1 and α_2 simultaneously at every

iteration. We obtain

$$
\begin{aligned}
O(\alpha_1, \alpha_2) \;&=\; \frac{1}{2}\alpha_1^2\, Q_{11} \;+\; \frac{1}{2}\alpha_2^2\, Q_{22} \;+\; \alpha_1\, \alpha_2\, Q_{12} \\
&+ \alpha_1 \sum_{j\neq 1,2} \alpha_j\, Q_{1j} \;+\; \alpha_2 \sum_{j\neq 1,2} \alpha_j\, Q_{2j} \;-\; \alpha_1\, l_1 \;-\; \alpha_2\, l_2 \\
&\stackrel{!}{=} \min
\end{aligned}
\tag{64}
$$

under the constraint that $-C \;\leq \alpha_{1,2} \;\leq\; C$. We first calculate the unconstrained minimum of eq. (65). If the corresponding values α_1 or α_2 violate the constraints, the values are corrected and set to the proper values (see below). If we set the derivatives of O in eq. (65) with respect to both α_1 and α_2 to zero, we obtain the linear equations

$$
\begin{aligned}
\alpha_1\, Q_{11} \;+\; \alpha_2\, Q_{12} \;+\; \sum_{j\neq 1,2} \alpha_j\, Q_{1j} \;-\; l_1 \;&=\; 0\,, \\
\alpha_1\, Q_{21} \;+\; \alpha_2\, Q_{22} \;+\; \sum_{j\neq 1,2} \alpha_j\, Q_{2j} \;-\; l_2 \;&=\; 0\,.
\end{aligned}
\tag{65}
$$

The linear equations are solved by

$$
\begin{aligned}
\alpha_1^{\rm new} \;&=\; \frac{-\, Q_{22} \left(\sum_{j\neq 1,2} \alpha_j\, Q_{1j} \;-\; l_1\right) \;+\; Q_{12} \left(\sum_{j\neq 1,2} \alpha_j\, Q_{2j} \;-\; l_2\right)}{Q_{11}\, Q_{22} \;-\; Q_{12}^2}\,, \\
\alpha_2^{\rm new} \;&=\; \frac{Q_{12} \left(\sum_{j\neq 1,2} \alpha_j\, Q_{1j} \;-\; l_1\right) \;-\; Q_{22} \left(\sum_{j\neq 1,2} \alpha_j\, Q_{2j} \;-\; l_2\right)}{Q_{11}\, Q_{22} \;-\; Q_{12}^2}\,,
\end{aligned}
\tag{66}
$$

which can be rewritten using $\sum_{j\neq 1,2} \alpha_j\, Q_{1j} \;-\; l_1 \;=\; F_1 \;-\; \alpha_1^{\rm old} Q_{11} \;-\; \alpha_2^{\rm old}\, Q_{12}$ and $\sum_{j\neq 1,2} \alpha_j\, Q_{2j} \;-\; l_2 \;=\; F_2 \;-\; \alpha_1^{\rm old} Q_{12} \;-\; \alpha_2^{\rm old}\, Q_{22}$ as

$$
\begin{aligned}
\alpha_1^{\rm new} \;&=\; \alpha_1^{\rm old} \;+\; \frac{F_2\, Q_{12} \;-\; F_1\, Q_{11}}{Q_{11}\, Q_{22} \;-\; Q_{12}^2}\,, \\
\alpha_2^{\rm new} \;&=\; \alpha_2^{\rm old} \;+\; \frac{F_1\, Q_{12} \;-\; F_2\, Q_{22}}{Q_{11}\, Q_{22} \;-\; Q_{12}^2}\,.
\end{aligned}
\tag{67}
$$

Note that if $\boldsymbol{K}$ is normalized, $\sum_{i=1}^{L} K_{ij}^2 \;=\; L$ (eqs. (23)), then $Q_{jj} \;=\; L$.

The minimum $(\alpha_1^{\rm new}, \alpha_2^{\rm new})$ has now to be checked against the "box constraints" $-C \;\leq \alpha_{1,2} \;\leq\; C$ and the values have to be properly corrected. The location of the unconstrained optimum can appear in six non-trivially different configurations relative to the position of the box. Fig. 9 shows these configurations for the left and upper border of the constraining box. The unconstrained minimum can be located in the box, $-C \;\leq \alpha_{1,2} \;\leq\; C$ (A), beside the box, $-C \;\leq \alpha_2 \;\leq\; C$ and $\alpha_1 \;<\; -C$ (B), above the box, $-C \;\leq \alpha_1 \;\leq\; C$ and $\alpha_2 \;>\; C$ (C), or in the upper left quadrant outside the box, $\alpha_1 \;<\; -C$ and $\alpha_2 \;>\; C$ (D,E,F). The latter case must be divided into three different cases,

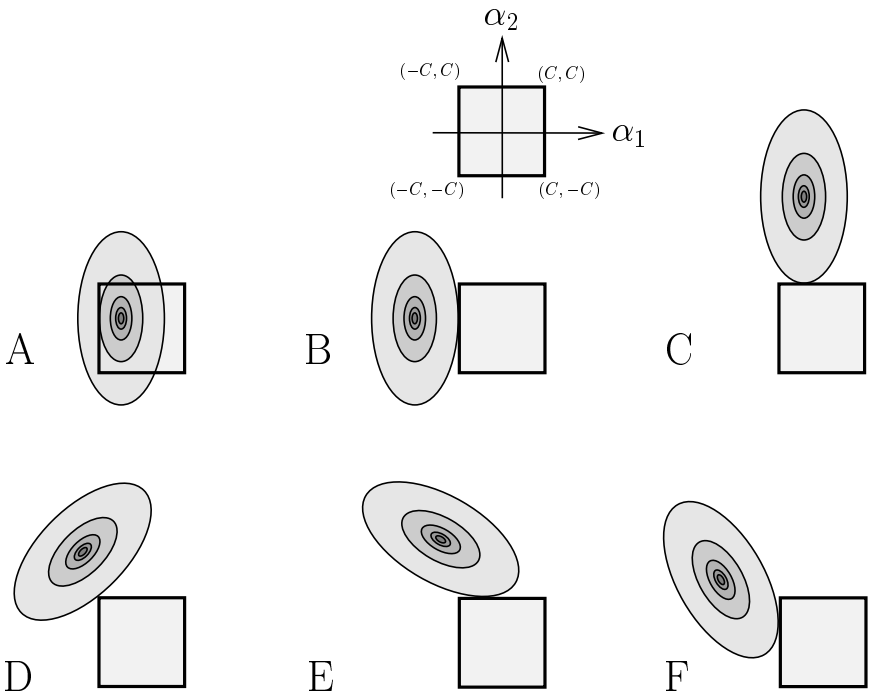

Figure 9: Illustration of the six non-trivial configurations of the unconstrained minimum of the objective function relative to the constraints on α_1 and α_2 indicated by the box. The minimum of the objective function is located in the center of the ellipsoids which indicate lines of equal value of the objective function. Similar configurations exist for the other three corners of the box.

because the objective function may take its minimal value at the upper left corner (D), at the upper border (E), or at the left border (F).

For case (A) the minimum is given by $(\alpha_1^{\text{new}}, \alpha_2^{\text{new}})$. For cases (B) and (C) the minimum can be determined by setting $\alpha_1^{\text{new}} = -C$ (B) or $\alpha_2^{\text{new}} = C$ (C), then updating the corresponding $F_{1,2}$ with the new value, and then solving eq. (64) for $\alpha_j = \alpha_2$ which gives α_2^{new} (B) or $\alpha_j = \alpha_1$ which gives α_1^{new} (C). In order to distinguish between the cases (D), (E), and (F), the derivatives of the objective function with respect to α_1 and α_2 are calculated for $\alpha_1 = -C$ and $\alpha_2 = C$ (upper left box corner). A positive derivative with respect to α_1 and a negative derivative with respect to α_2 indicate case (D), a negative derivative with respect to α_1 and a negative derivative with respect to α_2 indicate case (E), a positive derivative with respect to α_1 and a positive derivative with respect to α_2 indicate case (F). For case (D) the optimal value is given by $\alpha_1 = -C$ and $\alpha_2 = C$ while the cases (E) and (F) are treated similar to cases (B) or (C).

B.2 Optimization Step for Regularization with Slack Variables and Correlation Threshold

The dual P-SVM optimization problem can be rewritten as

$$\begin{aligned} \min_{\binom{\boldsymbol{\alpha}^+}{\boldsymbol{\alpha}^-}} \quad & \frac{1}{2}\begin{pmatrix}\boldsymbol{\alpha}^+\\ \boldsymbol{\alpha}^-\end{pmatrix}^\top \begin{pmatrix} \boldsymbol{K}^\top\boldsymbol{K} & -\boldsymbol{K}^\top\boldsymbol{K} \\ -\boldsymbol{K}^\top\boldsymbol{K} & \boldsymbol{K}^\top\boldsymbol{K}\end{pmatrix}\begin{pmatrix}\boldsymbol{\alpha}^+\\ \boldsymbol{\alpha}^-\end{pmatrix} \\ & - \boldsymbol{y}^\top\ (\boldsymbol{K}\ \ -\boldsymbol{K})\begin{pmatrix}\boldsymbol{\alpha}^+\\ \boldsymbol{\alpha}^-\end{pmatrix} + \epsilon\begin{pmatrix}\mathbf{1}\\ \mathbf{1}\end{pmatrix}^\top\begin{pmatrix}\boldsymbol{\alpha}^+\\ \boldsymbol{\alpha}^-\end{pmatrix} \\ \text{s.t.} \quad & \mathbf{0} \le \begin{pmatrix}\boldsymbol{\alpha}^+\\ \boldsymbol{\alpha}^-\end{pmatrix} \le C\begin{pmatrix}\mathbf{1}\\ \mathbf{1}\end{pmatrix}\ . \end{aligned} \tag{68}$$

$\binom{\boldsymbol{a}}{\boldsymbol{b}}$ denotes a column vector combining the elements of $\boldsymbol{a}$ and $\boldsymbol{b}$. If we set

$$\begin{aligned} \tilde{\boldsymbol{\alpha}} &:= \begin{pmatrix}\boldsymbol{\alpha}^+\\ \boldsymbol{\alpha}^-\end{pmatrix}\ , \\ \tilde{\boldsymbol{Q}} &:= \begin{pmatrix} \boldsymbol{K}^\top\boldsymbol{K} & -\boldsymbol{K}^\top\boldsymbol{K} \\ -\boldsymbol{K}^\top\boldsymbol{K} & \boldsymbol{K}^\top\boldsymbol{K}\end{pmatrix}\ , \text{ and} \\ \tilde{l} &= \boldsymbol{y}^\top\ (\boldsymbol{K}\ -\ \boldsymbol{K}) - \epsilon\begin{pmatrix}\mathbf{1}\\ \mathbf{1}\end{pmatrix}^\top\ , \end{aligned} \tag{69}$$

then we obtain the optimization setting from Section B.1, eqs. (61), with the lower bound equal to 0 instead of $-C$. This optimization problem can be solved by the SMO method of Section B.1, however, the dimensionality of the parameter vector has doubled.

Fortunately, above optimization problem can be formulated in compact form if we exploit the facts that for $i \le P$ we obtain $\tilde{F}_i = F_j^+$, $\tilde{F}_{(P+i)} = F_j^-$, and $\tilde{Q}_{ij} = -\tilde{Q}_{(P+i)j}$ for P complex features. From the facts presented at the beginning of Appendix B we know: $\tilde{\alpha}_i \cdot \tilde{\alpha}_{(P+i)} = 0$ and $\tilde{F}_i = -\tilde{F}_{(P+i)} + 2\,\epsilon$, which leads to an efficient implementation of the SMO optimization step. We find that $\tilde{\alpha}_i$ changes only if $\tilde{\alpha}_{(P+i)} = 0$ and vice versa, and that both $\tilde{F}_i$ and $\tilde{F}_{(P+i)}$ can be stored in the same variable. That means we must consider either $\tilde{\alpha}_i$ or $\tilde{\alpha}_{(P+i)}$ during optimization. Only for $\tilde{\alpha}_i = \tilde{\alpha}_{(P+i)} = 0$ both variables must be checked for updating which also is very efficient. To perform this check early in the optimization procedure and, therefore, to determine early if $\tilde{\alpha}_i > 0$ or $\tilde{\alpha}_{(P+i)} > 0$, initialization of the SMO procedure with $\alpha_j = 0$ is to be preferred (besides the positive effect that this initialization starts with the most sparse solution).

In conclusion, the implementation of the SMO technique for the correlation threshold regularization scheme is almost as efficient as for the slack variable regularization scheme treated in Section B.1. Only if $\tilde{\alpha}_i = \tilde{\alpha}_{(P+i)} = 0$, computational overhead is required.

B.3 Choice of Variables

Now we turn to the problem how to select the pair of variables for the next SMO iteration. The first variable, α_1, is chosen for an equation, where the Karush-Kuhn-Tucker (KKT) conditions are not fulfilled. If no such variable exists then the optimum has been found.

From the consideration before Section B.1 it follows that $\alpha_j^+ = \boldsymbol{C}$ implies $\xi_j^+ \geq 0$. This, together with the considerations at the beginning of Section B, let us deduce that the KKT conditions are met if

$$\begin{array}{rcl} \alpha_j^+ = 0 & \Longrightarrow & F_j^+ \geq 0 \\ 0 < \alpha_j^+ < C & \Longrightarrow & F_j^+ = 0 \\ \alpha_j^+ = C & \Longrightarrow & F_j^+ \leq 0 \end{array} \tag{70}$$

for α_j^+ and

$$\begin{array}{rcl} \alpha_j^- = 0 & \Longrightarrow & F_j^- \geq 0 \\ 0 < \alpha_j^- < C & \Longrightarrow & F_j^- = 0 \\ \alpha_j^- = C & \Longrightarrow & F_j^- \leq 0 \end{array} \tag{71}$$

for α_j^-. In order to find a proper α_1 we must check whether conditions eqs. (71) and eqs. (72) are fulfilled for the dual variables $\alpha_j^\pm$, for which $0 < \alpha_j^\pm < C$. If this is not the case, on-bound variables are checked ($\alpha_j^\pm = 0$ or $\alpha_j^\pm = C$). The evaluation of eqs. (71) and eqs. (72) does not increase the computational costs because the F_j are updated according to the new α_1 and α_2 at each SMO iteration.

After the choice of α_1 we check all variables (except α_1) and compute the optimal update according to Section B.1. The second variable, α_2, is chosen such that $\left|\alpha_{1,2}^{\text{new}} - \alpha_{1,2}^{\text{old}}\right|$ is maximal for α_1 or α_2. In contrast to Platt's SMO variant we do not use an approximation for the update values. It turned out that for matrices with small eigenvalues this gives a considerable speed up and the loss of speed for computing the exact update values is small.

For leave-one-out cross validation successive P-SVM optimization problems are similar to each other because only one training data point is exchanged. Therefore, support vectors of previous optimizations can be marked, i.e. "primed", and checked first in the new optimization problem for choosing α_1 and α_2. The speed up with priming is due to the fact that large updates are done earlier and following updates are more precise. We recommend to initialize the SMO with $\boldsymbol{\alpha} = \mathbf{0}$, because then the matrix $\boldsymbol{Q}$ must not be computed completely in order to compute the F_j, but has only to be evaluated at positions where $\alpha_j \neq 0$.

C Measurements, Kernels, and Dot Products

In this section we address the question under what conditions a "measurement kernel" which gives rise to a measured matrix $\boldsymbol{K}$ can be interpreted as a dot product between the "row" and "column" objects of a "matrix data" set. Section C.1 treats the case, where "row" and "column" objects are from different sets. We will show that under mild conditions the kernel corresponds to a dot product between feature vectors which are assigned to the objects and which live in a Hilbert space, where the dot product always exists for finite and almost always exists for infinite many "row" objects. The classification or regression function, which is chosen by the P-SVM, exists for all "column" objects. Section C.2 treats the case of pairwise data. We obtain results similar to Section C.1 but we will construct a classification or regression function in a Minkowski space.

C.1 Matrix Data

Let us assume that "column" objects x ("samples") and "row objects" z ("complex features") are from sets $\mathcal{X}$ and $\mathcal{Z}$, which can both be completed by a σ-algebra and a measure μ to a measurable spaces. We construct Hilbert spaces on these sets, but need some definitions first.

Let $(\mathcal{U}, \mathbb{B}, \mu)$ be a measurable space with σ-algebra $\mathbb{B}$ and a σ-additive measure μ on the set $\mathcal{U}$. We consider functions $f\colon \mathcal{U} \to \mathbb{R}$ on the set $\mathcal{U}$. A function f is called μ-measurable on $(\mathcal{U}, \mathbb{B})$ if $f^{-1}([a,b]) \in \mathbb{B}$ for all $a, b \in \mathbb{R}$, and μ-integrable if $\int_{\mathcal{U}} f \, d\mu < \infty$. We define

$$\|f\|_{L^2_\mu} := \left(\int_{\mathcal{U}} f^2 \, d\mu \right)^{\frac{1}{2}} \tag{72}$$

and the set

$$L^2_\mu(\mathcal{U}) := \left\{ f : \mathcal{U} \to \mathbb{R}; f \text{ is } \mu\text{-measurable and } \|f\|_{L^2_\mu} < \infty \right\} . \tag{73}$$

$L^2_\mu(\mathcal{U})$ is a Banach space with norm $\|\cdot\|_{L^2_\mu}$. If we define the dot product

$$\langle f, g \rangle_{L^2_\mu(\mathcal{U})} := \int_{\mathcal{U}} f \; g \; d\mu \tag{74}$$

then the Banach space $L^2_\mu(\mathcal{U})$ is a Hilbert space with a dot product $\langle \cdot, \cdot \rangle_{L^2_\mu(\mathcal{U})}$. For simplicity, we denote this Hilbert space by $L^2(\mathcal{U})$. $L^2(\mathcal{U}_1, \mathcal{U}_2)$ is the Hilbert space of functions k with $\int_{\mathcal{U}_1} \int_{\mathcal{U}_2} k^2(\boldsymbol{u}_1, \boldsymbol{u}_2) \; d\mu(u_2) \; d\mu(u_1) < \infty$ using the product measure of $\mu(U_1 \times U_2) = \mu(U_1)\mu(U_2)$. Let ℓ^2 be the Hilbert space of the set of infinite vectors $\boldsymbol{a} = (a_1, a_2, \dots)$ where $\sum_i a_i^2$ converges which possesses the dot product $\langle \boldsymbol{a}, \boldsymbol{b} \rangle_{\ell^2} = \sum_i a_i b_i$ and the norm $\|\boldsymbol{a}\|_{\ell^2} = \left(\sum_i a_i^2\right)^{\frac{1}{2}}$. With these definitions we see that $H_1 := L^2(\mathcal{Z})$, $H_2 := L^2(\mathcal{X})$, and $H_3 := L^2(\mathcal{X}, \mathcal{Z})$ are Hilbert spaces of L^2-functions with domains $\mathcal{X}$, $\mathcal{Z}$, and $\mathcal{X} \times \mathcal{Z}$, respectively. The dot product in H_i is denoted by $\langle \cdot, \cdot \rangle_{H_i}$.

Let us now assume that $k \in H_3$. k induces a Hilbert-Schmidt operator T_k:

$$f(x) = (T_k\alpha)(x) = \int_{\mathcal{Z}} k(x,z)\ \alpha(z)\ d\mu(z)\ , \tag{75}$$

which maps $\alpha \in H_1$ (a parameterization) to $f \in H_2$ (a classifier).

If we set $\mu(z) = \sum_{j=1}^{P} \delta\left(z^j\right)$, we recover the P-SVM classification function (without b), eq. (33), with $\alpha_j = \alpha(z^j)$

$$f(u) = \sum_{j=1}^{P} \alpha_j\ k\left(u, z^j\right) = \sum_{j=1}^{P} \alpha_j\ K_{(u)j}\ . \tag{76}$$

Here $\delta\left(z^j\right)$ is the Dirac delta function at location z^j. Note, that sums of Dirac functions define a measure (see Werner, 2000, page 464, example (c)).

We will now prove that a kernel k is a dot product for almost all pairs of (x, z) in some space if

(1) "column" objects ("samples") x are from a set $\mathcal{X}$ which can be completed to a measurable space,

(2) "row" objects ("complex features") z are from a set $\mathcal{Z}$ which can be completed to a measurable space, and

(3) the kernel k is from $L^2(\mathcal{X}, \mathcal{Z})$.

If $\int_{\mathcal{Z}}(k(x,z))^2\ \ d\mu\,(z) \leq K^2$ then the space, where k evaluates a dot product, can be identified as ℓ^2. Further, the regression or classification function f is continuous and the expansion in orthonormal functions converges absolutely and uniformly. The kernel k can be interpreted as mapping two objects, a "column" object x and "row" object z into a common space. In contrast to Mercer kernels the kernel k defines *two* mappings into the feature or measurement space. Fig. 10 depicts the situation: "column" objects (circles) and "row" objects (squares) are both mapped into a common space. In the measurement space the "column" objects are used to describe the normal vector of the separating hyperplane.

The next theorem provides assumptions for a kernel computing a dot product between the object's feature vectors.

Theorem 1 (Singular Value Expansion)
Let α be from H_1 and let k be a kernel from H_3 which defines a Hilbert-Schmidt operator $T_k : H_1 \to H_2$

$$\left(T_k\alpha\right)(x) = f(x) = \int_{\mathcal{Z}} k(x,z)\ \alpha(z)\ dz\ . \tag{77}$$

Then $\|f\|_{H_2}^2 = \langle T_k^* T_k \alpha, \alpha\rangle_{H_1}$, where T_k^* is the adjoint operator of T_k, and there exists an expansion

$$k(x,z) = \sum_n s_n\ e_n(z)\ g_n(x) \tag{78}$$

which converges in the L^2-sense. The $s_n \geq 0$ are the singular values of T_k, and $e_n \in H_1$, $g_n \in H_2$ are the corresponding orthonormal functions.

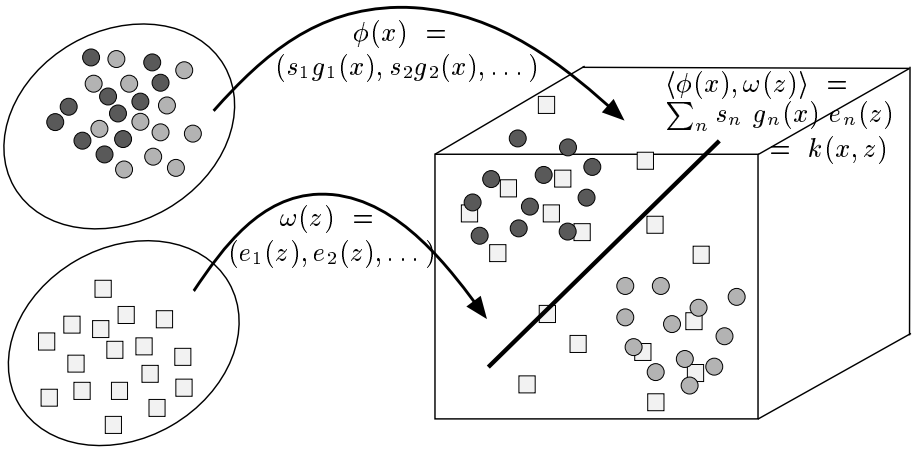

Figure 10: Interpretation of the measurement kernel applied to a set of "column" objects (circles) which should be classified and a set of "row" objects (squares) used to describe the "column" objects. Evaluating the kernel $k(x, z)$ is equivalent to first mapping both objects x and z into a common vector space (right) and then performing a scalar product (cf. eq. (79)). Dark and light circles indicate class membership, and the classification boundary (black line) is described by the "support row objects".

Proof.
From $f = T_k\alpha$ we obtain

$$\|f\|_{H_2}^2 \;=\; \langle T_k\alpha, T_k\alpha\rangle_{H_2} \;=\; \langle T_k^* T_k\alpha, \alpha\rangle_{H_1} \;. \tag{79}$$

The singular value expansion of T_k is

$$T_k\alpha \;=\; \sum_n s_n \langle \alpha, e_n\rangle_{H_1} \; g_n \tag{80}$$

(see Werner, 2000, Theorem VI.3.6). The values s_n are the singular values for the orthonormal systems $\{e_n\}$ on H_1 and $\{g_n\}$ on H_2. We define $r_{nm} := \langle T_k e_n, g_m\rangle_{H_2}$, where the sum

$$\sum_m r_{nm}^2 \;=\; \sum_m \left(\langle T_k e_n, g_m\rangle_{H_2}\right)^2 \;\le\; \|T_k e_n\|_{H_2}^2 \;<\; \infty \tag{81}$$

converges because of Bessel's inequality (the $\le$-sign). Next we complete the orthonormal system (ONS) $\{e_n\}$ to an orthonormal basis (ONB) $\{\tilde{e}_l\}$ by adding an ONB of the kernel $\ker(T_k)$ of the operator T_k to the ONS $\{e_n\}$. The function $\alpha \in H_1$ possesses an unique representation through this basis: $\alpha = \sum_l \langle \alpha, \tilde{e}_l\rangle_{H_1} \; \tilde{e}_l$. We obtain

$$T_k\alpha \;=\; \sum_l \langle \alpha, \tilde{e}_l\rangle_{H_1} \; T_k\tilde{e}_l \;. \tag{82}$$

Because $T_k\tilde{e}_l = 0$ for all $\tilde{e}_l \in \ker(T_k)$, the image $T_k\alpha$ can be expressed through the ONS $\{e_n\}$:

$$T_k\alpha = \sum_n \langle \alpha, e_n\rangle_{H_1}\ T_k e_n \tag{83}$$

$$= \sum_n \langle \alpha, e_n\rangle_{H_1}\ \left(\sum_m \langle T_k e_n, g_m\rangle_{H_2}\ g_m\right) = \sum_{n,m} r_{nm}\langle \alpha, e_n\rangle_{H_1}\ g_m\ .$$

Here we used the fact that $\{g_m\}$ is an ONB of the range of T_k and, therefore, $T_k e_n = \sum_m \langle T_k e_n, g_m\rangle_{H_2}\ g_m$.
Because the set of functions $\{e_n(z)\ g_m(x)\}$ are an ONS in H_3 (which can be completed to an ONB) and $\sum_{n,m} r_{nm}^2 < \infty$ (cf. eq. (82)), the kernel

$$\tilde{k}(z,x) := \sum_{n,m} r_{nm}\ e_n(z)\ g_m(x) \tag{84}$$

is from H_3. We observe that the induced Hilbert-Schmidt operator $T_{\tilde{k}}$ is equal to T_k which can be seen with eq. (84):

$$\left(T_{\tilde{k}}\alpha\right)(x) = \sum_{n,m} r_{nm}\langle \alpha, e_n\rangle_{H_1}\ g_m(x) = (T_k\alpha)(x)\ . \tag{85}$$

It follows that the kernel k and kernel $\tilde{k}$ are equal except for a set with zero measure, i.e. $k =_\mu \tilde{k}$. We obtain $\langle T_k e_l, g_t\rangle_{H_1} = \delta_{lt}\ s_l$ from the singular value decomposition eq. (81) and $\langle T_k e_l, g_t\rangle_{H_1} = r_{lt}$ from eq. (86), and, therefore, $r_{lt} = \delta_{lt} s_l$. Inserting $r_{nm} = \delta_{nm} s_n$ into eq. (85) proves the theorem.
■
As a consequence of this theorem we can define a mapping $\boldsymbol{\omega}$ of "row" objects z and a mapping $\boldsymbol{\phi}$ "column" objects x into a common feature space where k is a dot product.

$$\begin{aligned} \boldsymbol{\phi}(x) &:= (s_1 g_1(x), s_2 g_2(x), \dots)\ , \\ \boldsymbol{\omega}(z) &:= (e_1(z), e_2(z), \dots)\ , \\ \langle \boldsymbol{\phi}(x), \boldsymbol{\omega}(z)\rangle &= \sum_n s_n\ e_n(z)\ g_n(x) = k(z,x)\ . \end{aligned} \tag{86}$$

For the classification case these mappings are depicted in Fig. 10. In this common space a hyperplane which separates the "column" objects with respect to the class label should be constructed, and it is solely described by the "row" objects or, equivalently, through directions in the common space. From eq. (84) we obtain for the classification or regression function

$$f(x) = \sum_n s_n\ \langle \alpha, e_n\rangle_{H_1}\ g_n(x)\ . \tag{87}$$

The classification or regression function is well defined because sets of zero measure vanish through integration in eq. (76), which is confirmed through expansion eq. (88), where the zero measure is "absorbed" in the terms $\langle \alpha, e_n\rangle_{H_1}$.

However, the kernel expansion and the dot product expansion is not ensured to converge absolutely and uniformly in x which is desired to exchange summation with integration or differentiation. Therefore, the expansion of the classification or regression function $f(x)$ into the ONS g_m (cf. eq. (88)) should be ensured to converge absolutely and uniformly in x to justify the analysis in eq. (8) and eq. (9). More importantly, absolute and uniform convergence of the sum eq. (88) implies that $f(x)$ is continuous as a function of x. This can be seen because e_n are eigenfunctions of the compact, positive, self-adjoint operator $(T_k^* \, T_k)^{\frac{1}{2}}$ and g_n are isometric images of e_n (see Werner, 2000, Theorem VI.3.6 and Text before Theorem VI.4.2). Hence, the orthonormal functions g_n are continuous.

To obtain absolute and uniform convergence of the sum for $f(x)$, we must enforce $\|k(x,\cdot)\|_{H_1}^2 \; \leq \; K^2$ as can be seen in the following corollary.

Corollary 1 (Linear Classification in ℓ^2)
Let the assumptions of Theorem 1 hold and let $\int_{\mathcal{Z}} (k(x,z))^2 \;\; dz \; \leq \; K^2$ for all $x \; \in \; \mathcal{X}$.
We define $\boldsymbol{w} := (\langle \alpha, e_1 \rangle_{H_1}, \langle \alpha, e_2 \rangle_{H_1}, \dots)$, and $\boldsymbol{\phi}(x) := (s_1 g_1(x), s_2 g_2(x), \dots)$. Then $\boldsymbol{w}, \boldsymbol{\phi}(x) \in \ell^2$, where $\|w\|_{\ell^2}^2 \; \leq \; \|\alpha\|_{H_1}^2$ and $\|\boldsymbol{\phi}(x)\|_{\ell^2}^2 \; \leq \; K^2$, and the following sum convergences absolutely and uniformly:

$$f(x) \; = \; \langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle_{\ell^2} \; = \; \sum_n s_n \; \langle \alpha, e_n \rangle_{H_1} \; g_n(x) \; . \tag{88}$$

Proof.
First we show that $\boldsymbol{\phi}(x) \in \ell^2$:

$$\begin{aligned} \|\boldsymbol{\phi}(x)\|_{\ell^2}^2 \; &= \; \sum_n \left(s_n \; g_n(x)\right)^2 \; = \; \sum_n \left((T_k e_n)(x)\right)^2 \qquad (89) \\ &= \; \sum_n \left(\langle k(x,.), e_n \rangle_{H_1}\right)^2 \; \leq \; \|k(x,.)\|_{H_1}^2 \\ &\leq \; \sup_{x \, \in \, \mathcal{X}} \{ \int_{\mathcal{Z}} (k(x,z))^2 \;\; dz \} \; \leq \; K^2 \; , \end{aligned}$$

where we used Bessel's inequality for the first "$\leq$", we used the supremum over $x \in \mathcal{X}$ for the second "$\leq$" (the supremum exists because $\{\int (k(x,z))^2 \;\; dz\}$ is a bounded subset of $\mathbb{R}$), and we used the assumption of the corollary for the last "$\leq$". To prove $\|w\|_{\ell^2}^2 \; \leq \; \|\alpha\|_{H_1}^2$ we use again Bessel's inequality:

$$\|w\|_{\ell^2}^2 \; = \; \sum_n \left(\langle \alpha, e_n \rangle_{H_1}\right)^2 \; \leq \; \|\alpha\|_{H_1}^2 \; . \tag{90}$$

Finally, we prove that the sum

$$f(x) \; = \; \langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle_{\ell^2} \; = \; \sum_n s_n \; \langle \alpha, e_n \rangle_{H_1} \; g_n(x) \tag{91}$$

converges absolutely and uniformly. The fact that the sum convergences in the L^2 sense follows directly from the singular value expansion of Theorem 1. We

now chose an $m \in \mathbb{N}$ with

$$\sum_{n=m}^{\infty} \left(\langle \alpha, e_n \rangle_{H_1}\right)^2 \;\leq\; \left(\frac{\epsilon}{K}\right)^2 \tag{92}$$

for $\epsilon > 0$ (because of eq. (91) such an m exists), and we apply the Cauchy-Schwarz inequality

$$\begin{aligned}
&\sum_{n=m}^{\infty} |s_n \; \langle \alpha, e_n \rangle_{H_1} \; g_n(x)| \\
&\leq \left(\sum_{n=m}^{\infty} \left(s_n \; g_n(x)\right)^2\right)^{\frac{1}{2}} \left(\sum_{n=m}^{\infty} \left(\langle \alpha, e_n \rangle_{H_1}\right)^2\right)^{\frac{1}{2}} \\
&\leq K \; \frac{\epsilon}{K} \;=\; \epsilon \;,
\end{aligned}$$

where we used inequalities eqs. (90) and (93). Because m is independent of x, the convergence is absolutely and uniformly, too.

■

Eq. (76) or, equivalently, (89) is a linear classification or regression function in ℓ^2. We find that the expansion of the classifier f converges absolutely and uniformly and, therefore, that f is continuous.

In the following we show the connection to the P-SVM, where we use $\mu(x) = \sum_{i=1}^{L} \delta\left(x^i\right)$, $\mu(z) = \sum_{i=j}^{P} \delta\left(z^j\right)$, and $\alpha_j := \alpha\left(z^j\right)$. We obtain

$$\begin{aligned}
f(x) \;&=\; \sum_{j=1}^{P} \alpha_j \; k\left(x, z^j\right) \;=\; \left\langle \boldsymbol{\phi}\left(x\right), \sum_{j=1}^{P} \alpha_j \; \boldsymbol{\omega}\left(z^j\right) \right\rangle \;, \\
\boldsymbol{X} \;&=\; \left(\boldsymbol{\phi}\left(x^1\right), \boldsymbol{\phi}\left(x^2\right), \ldots, \boldsymbol{\phi}\left(x^L\right)\right) \;, \\
\boldsymbol{Z} \;&=\; \left(\boldsymbol{\omega}\left(z^1\right), \boldsymbol{\omega}\left(z^2\right), \ldots, \boldsymbol{\omega}\left(z^P\right)\right) \;, \\
\boldsymbol{w} \;&=\; \sum_{j=1}^{P} \alpha_i \; \boldsymbol{\omega}\left(z^j\right) \text{ (expansion into support vectors)}, \\
K_{ij} \;&=\; \left\langle \boldsymbol{\phi}\left(x^i\right), \boldsymbol{\omega}\left(z^j\right) \right\rangle \;=\; \sum_n s_n \; e_n\left(z^j\right) \; g_n\left(x^i\right) \;=\; k\left(x^i, z^j\right), \\
\boldsymbol{K} \;&=\; \boldsymbol{X}^\top \; \boldsymbol{Z} \;, \text{ and} \\
\|f\|_{H_2}^2 \;&=\; \boldsymbol{\alpha}^\top \boldsymbol{K}^\top \boldsymbol{K} \; \boldsymbol{\alpha} \;=\; \|\boldsymbol{X}^\top \; \boldsymbol{w}\|_2^2 \text{ (the objective function).}
\end{aligned} \tag{93}$$

At the end of Section 2.5 we mentioned that $\boldsymbol{w}$ is not unique with respect to the subspace which is mapped to zero by the matrix $\boldsymbol{X}$. Here we obtain an analog result: $\boldsymbol{w}$ is not unique with respect to the subspace which is mapped to the zero function by T_k, that is components of α which are in the subspace which is mapped to the zero function by T_k have no impact on $\boldsymbol{w}$. Interestingly, we recovered the new objective function eq. (6) as the L^2-norm $\|f\|_{H_2}^2$ on the classification function. This, again, motivates the use of the new objective function as a capacity measure.

We found that the primal problem of the P-SVM (e.g. eq. (27)) corresponds to the formulation in H_2, while the dual (e.g. eq. (32)) corresponds to the formulation in H_1. Primal and dual P-SVM formulations can be transferred into each other via the property $\langle T_k\alpha, T_k\alpha\rangle_{H_2} = \langle T_k^* T_k\alpha, \alpha\rangle_{H_1}$.

The objective function eq. (6) minimizes the capacity of the classifier in H_2, that is the range of T_k. However, regularization schemes restrict the domain of T_k via a maximum norm on the function α: $\max_x |\alpha(x)| \leq C$ (classification and regression using slack variables) or minimize the domain capacity by pushing the functions towards the zero function via minimizing $\|\alpha\|_1 = \int_{\mathcal{Z}} |\alpha(z)|\, d\mu(z)$ (feature selection using the correlation threshold). Note that $\delta(z^j)$ has measure larger than zero and, therefore, z^j never belongs to a zero measure set.

C.2 Kernels for Pairwise Data

Pairwise data is a special case of matrix data, for which "row" and "column" objects are from the same set. Therefore, only one mapping $\boldsymbol{\phi}$ into the feature space exists and an eigenvalue decomposition has to be performed instead of the expansion into singular values from the previous section. The consequence, however, is that that eigenvalues may become negative (see the following theorem).

Theorem 2 (Eigenvalue Expansion)
Let definitions and assumptions be as in Theorem 1. Let $H_1 = H_2 = H$ and let k be symmetric. Then there exists an expansion $k(x,z) = \sum_n \nu_n\, e_n(z)\, e_n(x)$ which converges in the L^2-sense. The ν_n are the eigenvalues of T_k with the corresponding orthonormal eigenfunctions e_n.

Proof.
This theorem is Theorem 87.7 in (Heuser, 1992).
■

If k is both continuous and positive definite and if H is compact, then the expansion for k in Theorem 2 converges uniformly and absolutely for all x (Mercer). As in previous section, we want for more general non-Mercer k the sum, which expands the classifier, to converge absolutely and uniformly (see following corollary).

Corollary 2 (Minkowski Space Classification)
Let the assumptions of Theorem 2 and $\int_{\mathcal{X}} (k(x,z))^2\, dz \leq K^2$, for all x, hold true. We define $\boldsymbol{w} := (\sqrt{|\nu_1|}\langle\alpha, e_1\rangle_H, \sqrt{|\nu_2|}\langle\alpha, e_2\rangle_H, \dots)$,
$\boldsymbol{\phi}(x) := (\sqrt{|\nu_1|}e_1(x), \sqrt{|\nu_2|}e_2(x), \dots)$, and denote by ℓ^2_S the space ℓ^2 with a given signature $S = (sign(\nu_1), sign(\nu_2), \dots)$. Then the following holds true:
(a) $\boldsymbol{w} \in \ell^2_S$ and $\|w\|^2_{\ell^2_S} = \langle T_k\alpha, \alpha\rangle_H$,
(b) If $\boldsymbol{\phi}(x) \in \ell^2_S$ then $\|\boldsymbol{\phi}(x)\|^2_{\ell^2_S} = k(x,x)$ in the L^2 sense, and
(c) the following sum convergences absolutely and uniformly:

$$f(x) = \langle \boldsymbol{w}, \boldsymbol{\phi}(x)\rangle_{\ell^2_S} = \sum_n \nu_n\, \langle\alpha, e_n\rangle_H\, e_n(x)\ . \tag{94}$$

Proof.
The fact that the sum which expands $f(x)$ convergences absolutely and uniformly is stated as Theorem (87.8) in (Heuser, 1992).
Next we want prove that $\boldsymbol{w} \in \ell_S^2$. The uniform convergence of $f(x)$ allows for

$$\begin{aligned} \infty \; > \; \langle \alpha, T_k \alpha \rangle_H \; &= \; \langle \alpha, f \rangle_H \; = \; \left\langle \alpha, \sum_n \nu_n \; \langle \alpha, e_n \rangle_H \; e_n \right\rangle_H \qquad (95) \\ &= \; \sum_n \nu_n \; \langle \alpha, e_n \rangle_H^2 \; = \; \|w\|_{\ell_S^2}^2 \; . \end{aligned}$$

Note, that the last equality is the definition of $\|\cdot\|_{\ell_S^2}^2$ (not $\|\cdot\|_H^2$).
If $\boldsymbol{\phi}(x) \in \ell_S^2$ then

$$\|\boldsymbol{\phi}(x)\|_{\ell_S^2}^2 \; = \; \sum_n \nu_n \; e_n(x)^2 \; = \; k(x,x) \qquad (96)$$

holds in the L^2 sense because of the eigenvalue expansion.
■
Eq. (95) is a linear classification or regression function in the Minkowski space ℓ_S^2. In comparison to Corollary 1 we have $\|w\|_{\ell_S^2}^2 \; = \; \boldsymbol{\alpha}^T \boldsymbol{K} \, \boldsymbol{\alpha}$ and we must assume that the expansion for $\|\boldsymbol{\phi}(x)\|_{\ell_S^2}^2$ does converge. As consequence only almost all classification tasks can be formulated in ℓ_S^2. However, the classification or regression function still converges always absolutely and uniformly.

References

P. Ahlgren, B. Jarneving, and R. Rousseau. Requirements for a cocitation similarity measure with special reference to pearson's correlation coefficient. *Journal of the American Society for Information Science and Technology*, 54: 550–560.

W. Bains and G. Smith. A novel method for nucleic acid sequence determination. *Journal of Theoretical Biology*, 135:303–307, 1988.

A. E. Bayer, J. C. Smart, and G. W. McLaughlin. Mapping intellectual structure of a scientific subfield through author cocitations. *Journal of the American Society for Information Science*, 41(6):444–452.

A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

A. Califano, G. Stolovitzky, and Y. Tu. Analysis of gene expression microarrays for phenotype classification. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 75–85, 1999.

W. Chu, S. S. Keerthi, and C. J. Ong. Bayesian support vector regression using a unified loss function. *IEEE Transactions on Neural Networks*, 2004. To appear.

T. Cremer, A. Kurz, R. Zirbel, S. Dietzel, B. Rinke, E. Schröck, M. R. Speichel, U. Mathieu, A. Jauch, P. Emmerich, H. Schertan, T. Ried, C. Cremer, and P. Lichter. Role of chromosome territories in the functional compartmentalization of the cell nucleus. *Cold Spring Harbor Symp. Quant. Biol.*, 58: 777–792, 1993.

R. Drmanac, I. Labat, I. Brukner, and R. Crkvenjakov. Sequencing of megabase plus DNA by hybridization: theory of the method. *Genomics*, 4:114–128, 1989.

L. Falquet, M. Pagni, P. Bucher, N. Hulo, C. J. Sigrist, K. Hofmann, and A. Bairoch. The PROSITE database, its status in 2002. *Nucleic Acids Research*, 30:235–238, 2002.

T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer. Classification on pairwise proximity data. In M. S. Kearns, S. A. Solla, and D. A. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11, pages 438–444. MIT Press, Cambridge, MA, 1999.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003. Special Issue on Variable and Feature Selection.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46(1-3):389–422, 2002.

R. Herbrich, T. Graepel, P. Bollmann-Sdorra, and K. Obermayer. Learning a preference relation for information retrieval. In *Working Notes of the AAAI Workshop on Information Retrieval'98*, pages 83–86, 1998.

H. Heuser. *Funktionalanalysis*. B. G. Teubner, Stuttgart, Germany, 3. edition, 1992.

L. J. Heyer, S. Kruglyak, and S. Yooseph. Exploring expression data: identification and analysis of coexpressed genes. *Genome Research*, 11:1106–1115, 1999.

S. Hochreiter, M. C. Mozer, and K. Obermayer. Coulomb classifiers: Generalizing support vector machines via an analogy to electrostatic systems. In S. Beckers, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 545–552. MIT Press, Cambridge, MA, 2003.

S. Hochreiter and K. Obermayer. Gene selection for microarray data. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology*. MIT Press, 2004a. To appear.

S. Hochreiter and K. Obermayer. Sphered support vector machine. Technical report, Technische Universität Berlin, Fakultät für Elektrotechnik und Informatik, 2004b.

T. Hofmann and J. M. Buhmann. Pairwise data clustering by deterministic annealing. *IEEE Transactions on Pattern Analysis and Machine Intelligence"*, 19(1):1–25, 1997.

M. G. Kendall and A. Stuart. *The advanced theory of statistics*. Charles Griffin & Co LTD, 4 edition, 1977.

J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the Association for Computing Machinery*, 46(5):604–632, 1999.

R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

D. Lipman and W. Pearson. Rapid and sensitive protein similarity searches. *Science*, 227:1435–1441, 1985.

Q. Lu, L. L. Wallrath, and S. C. R. Elgin. Nucleosome positioning and gene regulation. *Journal of Cellular Biochemistry*, 55:83–92, 1994.

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

Y. Lysov, V. Florent'ev, A. Khorlin, K. Khrapko, V. Shik, and A. Mirzabekov. DNA sequencing by hybridization with oligonucleotides. *Doklady Academy Nauk USSR*, 303:1508–1511, 1988.

O. L. Mangasarian. Generalized support vector machines. Technical Report 98-14, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, 1998.

C. B. Mazza, N. Sukumar, C. M. Breneman, and S. M. Cramer. Prediction of protein retention in ion-exchange systems using molecular descriptors obtained from crystal structure. *Anal. Chem.*, 73:5457–5461, 2001.

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.

S. L. Pomeroy, P. Tamayo, M. Gaasenbeek, L. M. Sturla, M. Angelo, M. E. McLaughlin, J. Y. H. Kim, L. C. Goumnerova, P. M. Black, C. Lau, J. C. Allen, D. Zagzag, J. M. Olson, T. Curran, C. Wetmore, J. A. Biegel, T. Poggio, S. Mukherjee, R. Rifkin, A. Califano, G. Stolovitzky, D. N. Louis, J. P. Mesirov, E. S. Lander, and T. R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870): 436–442, 2002.

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.

J. W. Scannell, C. Blakemore, and M. P. Young. Analysis of connectivity in the cat cerebral cortex. *The Journal of Neuroscience*, 15(2):1463–1483, 1995.

U. Scherf, D. T. Ross, M. Waltham, L. H. Smith, J. K. Lee, L. Tanabe, K. W. Kohn, W. C. Reinhold, T. G. Myers, D. T. Andrews, D. A. Scudiero, M. B. Eisen, E. A. Sausville, Y. Pommier, D. Botstein, P. O. Brown, and J. N. Weinstein. A gene expression database for the molecular pharmacology of cancer. *Nature Genetics*, 24(3):236–244, 2000.

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.

M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, R. C. T. Aguiar J. L. Kutok, M. Gaasenbeek, M. Angelo, M. Reich, T. S. Ray G. S. Pinkus, M. A. Koval, K. W. Last, A. Norton, J. Mesirov T. A. Lister, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub. Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1):68–74, 2002.

C. J. Sigrist, L. Cerutti, N. Hulo, A. Gattiker, L. Falquet, M. Pagni, A. Bairoch, and P. Bucher. PROSITE: A documented database using patterns and profiles as motif descriptors. *Brief Bioinformatics*, 3:265–274, 2002.

E. Southern. United Kingdom patent application GB8810400, 1988.

L. J. van't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.

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.

D. Werner. *Funktionalanalysis*. Springer-Verlag, Berlin, Germany, 3. edition, 2000.

J. Weston, S. Mukherjee, O. Chapelle, M. Pontil, T. Poggio, and V. Vapnik. Feature selection for SVMs. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, Cambridge, MA, 2000.

H. D. White and K. W. McCain. Bibliometrics. *Annual Review of Information Science and Technology*, 24:119–186.