Contents

Potential Support Vector Machines for Dyadic 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, where two sets of objects ("row" and "column" objects) are represented by a matrix of numerical values which describe their mutual relationships. The new technique, called "Potential Support Vector Machine" (P-SVM), is a large-margin based method for the construction of classifiers and regression functions for the "column" objects. Contrary to standard support vector machine approaches, the P-SVM minimizes a scale-invariant capacity measure under a new set of constraints. As a result, the P-SVM leads to a usually sparse expansion of the classification or regression functions in terms of the "row" rather than the "column" objects and can handle data matrices which are neither positive definite nor square. We then describe two complementary regularization schemes. The first scheme improves generalization performance for the classification and regression tasks, the second scheme leads to the selection of a small, informative set of "row" objects and can be applied to feature selection. Benchmarks are performed with toy as well as with several real world data sets, including data from the UCI repository, protein classification, web-page classification, and DNA microarray data, and cover classification, regression, and feature selection tasks.

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

	A	B	C	D
α	1.3	−2.2	−1.6	7.8
β	−1.8	−1.1	7.2	2.3
χ	1.2	1.9	−2.9	−2.2
δ	3.7	0.8	−0.6	2.5
ε	9.2	−9.4	−8.3	9.2
ϕ	−7.7	8.6	−9.7	−7.4
γ	−4.8	0.1	−1.2	0.9

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

	A	B	C	D	E	F	G
A	0.9	−0.1	−0.8	0.5	0.2	−0.5	−0.7
B	−0.1	0.9	0.6	0.3	−0.7	−0.6	0.3
C	−0.8	0.6	0.9	0.2	−0.6	0.6	0.5
D	0.5	0.3	0.2	0.9	0.7	0.1	0.3
E	0.2	−0.7	−0.6	0.7	0.9	−0.9	−0.5
F	−0.5	−0.6	0.6	0.1	−0.9	0.9	0.9
G	−0.7	0.3	0.5	0.3	−0.5	0.9	0.9

Figure 1: (a) Dyadic data: Column objects $\{A, B, C, D\}$ and row objects $\{\alpha, \beta, \ldots, \gamma\}$ are represented by a matrix of numerical values which describe their natural relationships. (b) Pairwise data: Special case where the set of row and column objects coincide.

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. 1a). "Column" objects are the objects to be described, while "row" objects are the objects which serve for their description. Then the whole dataset can be represented using a rectangular matrix whose entries denote the relationships between the corresponding "row" and the "column" objects. In the following we will call representations of this form *dyadic data*. If "row" and "column" objects are from the same set (Fig. 1b), the representation is usually called *pairwise data*, and the entries of the matrix can often be interpreted as the degree of similarity (or dissimilarity) between pairs of objects.

Dyadic descriptions are more powerful than vector-based descriptions, but vectorial data can always be brought into dyadic form, when required. This is often done for kernel-based classifiers or regression functions (Schölkopf and Smola, 2002; Vapnik, 1998), where a Gram matrix of mutual similarities (Fig. 1b) is calculated before the predictor is learned. A similar procedure can also be used in the case where the "row" and "column" objects are from different sets (Fig. 1a). 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. In many cases, however, dyadic descriptions emerge, because the matrix entries are measured directly.

Pairwise data representation can be found in many datasets which are generated by measuring similarities. Examples 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), co-citation matrices for text documents (White and McCain, 1989; Bayer et al., 1990; Ahlgren et al., 2003), or binary connectivity matrices which summarize the hyperlinks between web-pages (Kleinberg, 1999). In general, these measured matrices are symmetric but may not be positive definite, and even if they are for the training set, they may not remain positive definite, if new examples are included. Examples for genuine *dyadic data* are DNA microarray data (Southern, 1988; Lysov et al., 1988; Drmanac et al., 1989; Bains and Smith, 1988), where the "column" objects are tissue or cell-line samples, the "row" objects are genes, and every sample-gene pair is related by the expression level of this particular gene in this particular sample. Other examples 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 is then characterized by the number of directed hyperlinks from row to column, which gives rise to a rectangular matrix of ordinal values[1]. Other examples include (i) images ("column" objects), which can be described by the scalar values (matrix elements) obtained from average linear or non-linear filter responses ("row" objects) to an image, (ii) time-series, which 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, (iii) customers of a company can be described by their product preferences or by their transaction data, (iv) documents in a database can be described by word-frequencies, or (v) 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 an Euclidean vector space. Difficulties, however, arise when the features are heterogeneous, and apples and oranges must be compared. What theoretical arguments would, for example, justify, to treat the values of a set of TAE descriptors as coordinates of a vector in Euclidean space?

Classification and regression problems on dyadic data, have been mostly addressed within the feature vector framework (Graepel et al., 1999; Mangasarian, 1998), i.e. the "feature map" method (Schölkopf and Smola, 2002). An "non-vectorial" approach to pairwise data is to interpret the 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

[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.

indefinite (but symmetric) matrices two other 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. 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. In the other approach directions of negative eigenvalues are processed by just flipping the sign of these eigenvalues. All three approaches, however, lead to positive semidefinite matrices for training set relations, but do not ensure that positive semidefiniteness still holds, if a new test object must be included. A fourth 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. So far, no general method exists for learning classifiers or regression functions from data represented dyadicly.

Here 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 show that, under mild assumptions, pairwise measurements are sufficient to create a vector space endowed with a dot product into which the "row" and the "column" objects are mapped (cf. Section 2.5 and Appendix A). Using this mathematical argument as a justification, we then construct the classification or 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 dyadic data. The new method is called the potential support vector machine (P-SVM) and 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 very good 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, which is different from the standard expansion in terms of "column" objects. This opens up another important application domain: a sparse expansion is equivalent to feature selection (see Guyon and Elisseeff, 2003; Hochreiter and Obermayer, 2004b; Kohavi and John, 1997; Blum and Langley, 1997 for reviews on feature selection). An efficient implementation of the P-SVM requires a modified sequential minimal optimization procedure for learning. This method is described in (Hochreiter and Obermayer, 2004a).

2 The Potential Support Vector Machine

2.1 The Standard SVM Approach

Consider a set $\left\{x^i \mid 1 \leq i \leq L\right\} \subset \mathcal{X}$ of objects which are described by feature vectors $\boldsymbol{x}_\phi^i \in \mathbb{R}^N$ and which form a training set $X_\phi = \left\{\boldsymbol{x}_\phi^1, \ldots, \boldsymbol{x}_\phi^L\right\}$. The index "$\phi$" is introduced, because we will later assume that the vectors $\boldsymbol{x}_\phi^i$ are images in $\mathbb{R}^N$ of a map ϕ which is induced either by a kernel or by a measurement function (and utilizing a kernel trick)[2]. Assume for the moment a simple binary classification problem, where class membership is indicated by labels $y_i \in \{+1, -1\}$, and a set $\{\mathrm{sign}\,(f)\}$ of linear classifiers with

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

which are parameterized by the weight vector $\boldsymbol{w}$ and the offset b ($\langle \cdot, \cdot \rangle$ denotes the dot product). The classification boundaries are given by the hyperplanes $f(\boldsymbol{x}_\phi) = 0$, and the margin γ can be calculated according to

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

If the hyperplane is given in its "canonical form" (Vapnik, 1995), then we obtain $\gamma = \|\boldsymbol{w}\|_2^{-1}$.

Standard SVM-techniques select the "canonical" hyperplane with the largest margin under the constraint of correct classification on the training set:

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

If the training data are not linearly separable, a large margin is 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 (Vapnik, 1998). For the set of all linear classifiers defined on X_ϕ, 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. 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. In (Schölkopf et al., 1999) instead of a sphere, an ellipsoid is fitted to the data which more accurately bounds the generalization error.

[2] The following considerations also hold for $N \to \infty$, if a Hilbert space like ℓ^2 is considered.

2.2 The Advantage of Scale Invariance

Both the selection of a classifier using the maximum margin principle and the values obtained for the generalization error bounds 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 two dimensional

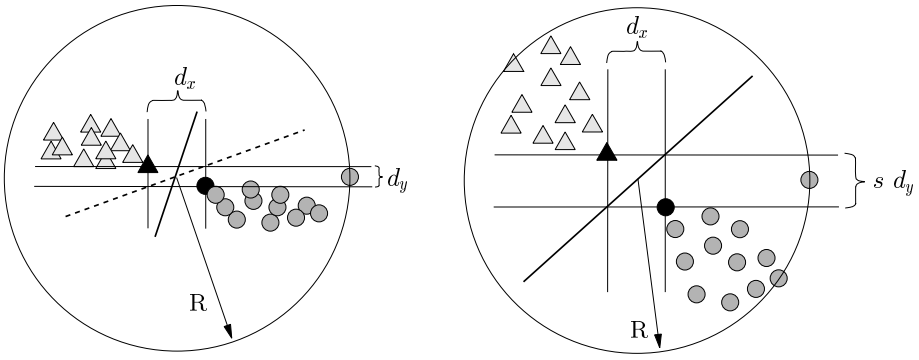

Figure 2: LEFT: data points from two classes (triangles and circles) are separated by the hyperplane with the largest margin (solid line). The two support vectors (black symbols) are separated by d_x along the horizontal and by d_y along the vertical axis, from which we obtain $\gamma = \frac{1}{2}\sqrt{d_x^2 + d_y^2}$ 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.

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. These considerations show, that 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. This situation may appear often in real data in higher dimensions because the situation is not present if in an n-dimensional space the $(n+1)$ border points are on a hypersphere. In all other situations scaling orthogonal to points on the hypersphere is possible. 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 to the normal vector $\boldsymbol{w}$ of the hyperplane with maximal margin γ are scaled to zero and $\tilde{\mathrm{R}} = \min_{t\in\mathbb{R}} \max_i |\left\langle \hat{\boldsymbol{w}}, \boldsymbol{x}_\phi^i \right\rangle + t| \leq \max_i |\left\langle \hat{\boldsymbol{w}}, \boldsymbol{x}_\phi^i \right\rangle|$, where $\hat{\boldsymbol{w}} := \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$. If the absolute value of t is small compared to the absolute values of $\left\langle \hat{\boldsymbol{w}}, \boldsymbol{x}_\phi^i \right\rangle$, 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:

$$\frac{\tilde{\mathrm{R}}^2}{\gamma^2} = \tilde{\mathrm{R}}^2 \|\boldsymbol{w}\|^2 \leq \max_i \langle \boldsymbol{w}, \boldsymbol{x}_\phi^i \rangle^2 \leq \sum_i \langle \boldsymbol{w}, \boldsymbol{x}_\phi^i \rangle^2 = \left\| \boldsymbol{X}_\phi^\top \boldsymbol{w} \right\|^2 , \quad (5)$$

where the matrix $\boldsymbol{X}_\phi := \left(\boldsymbol{x}_\phi^1, \boldsymbol{x}_\phi^2, \ldots, \boldsymbol{x}_\phi^L \right)$ contains all the training vectors $\boldsymbol{x}_\phi^i$. The second inequality is the squared bound on the maximum norm by the Euclidean norm. Its worst case factor is L, but the bound is tight (e.g. if only one component differs from zero).

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

$$\boldsymbol{w}^\top \boldsymbol{X}_\phi \, \boldsymbol{X}_\phi^\top \boldsymbol{w} = \left\| \boldsymbol{X}_\phi^\top \boldsymbol{w} \right\|^2 \quad (6)$$

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 are rotated towards directions of low variance of the data when compared with the standard maximum margin solution. To quantify the difference of using the new objective instead of the margin as in the standard SVM approach we used the breast-cancer data set from the UCI collection (see experiments in Subsection 3.1.1). We report in that sphering in feature space changes the classification bondary of a SVM-based classifier. We chose the first training set and compared the the SVM-base classifiers with the differen objective for different C-values for the RBF-kernel with $\sigma = 1$ and for differend σ with C-value of 2.0. For all pairs of objectives we computed the angle of the solution in feature space:

$$\phi = \arccos\left(\frac{\langle \boldsymbol{w}_{\mathrm{svm}}, \boldsymbol{w}_{\mathrm{sphered}} \rangle}{\sqrt{\langle \boldsymbol{w}_{\mathrm{svm}}, \boldsymbol{w}_{\mathrm{svm}} \rangle} \, \sqrt{\langle \boldsymbol{w}_{\mathrm{sphered}}, \boldsymbol{w}_{\mathrm{sphered}} \rangle}} \right) =$$

$$\arccos\left(\frac{\sum_{i,j} \alpha_i^{\mathrm{svm}} \alpha_j^{\mathrm{sphered}} \, y_i \, k\left(\boldsymbol{x}^i, \boldsymbol{x}^j\right)}{\sqrt{\sum_{i,j} \alpha_i^{\mathrm{svm}} \alpha_j^{\mathrm{svm}} \, y_i \, y_j \, k\left(\boldsymbol{x}^i, \boldsymbol{x}^j\right)} \, \sqrt{\sum_{i,j} \alpha_i^{\mathrm{sphered}} \alpha_j^{\mathrm{sphered}} \, k\left(\boldsymbol{x}^i, \boldsymbol{x}^j\right)}} \right) .$$

The results are given in Fig. 3 for C-values (left) and σ-values (right). The solution converge to each other if lower training error is enforces and more training examples are used but differ considerably if regularization is allowed.

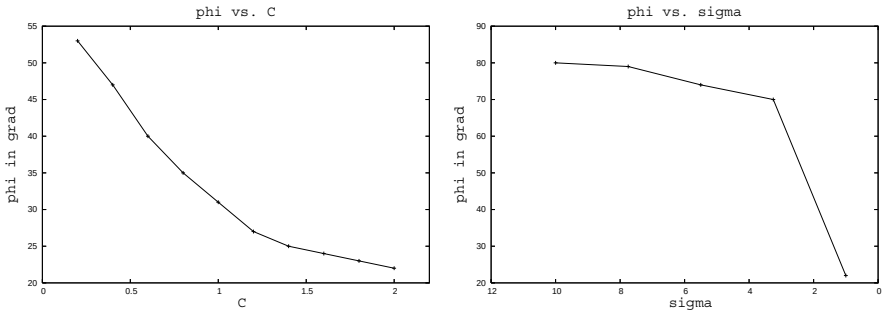

Figure 3: The angle between the SVM-based classifiers where one is the standard SVM solution and one sphering in feature space. The angle vs. the regularization value C (left) and vs. the kernel regularization parameter σ (right).

The new objective is well defined also for cases where $\boldsymbol{X}_\phi \, \boldsymbol{X}_\phi^\top$ or/and $\boldsymbol{X}_\phi^\top \, \boldsymbol{X}_\phi$ 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 $\boldsymbol{X}_\phi \, \boldsymbol{X}_\phi^\top = \mathbf{I}$ and we recover the classical SVM. Note, however, that whitening can easily be performed in input space but becomes nontrivial if the data is mapped to a high-dimensional feature space using a kernel function[3].

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 capacity measure for the model class like the margin is? It is, and in (Hochreiter and Obermayer, 2004c) 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.

2.3 Constraints for Complex Features

The next step is to formulate a set of constraints which (i) enforce a good performance on the training set and (ii) regard the conditions imposed by the new idea of treating dyadic data. We assume that "row" and "column" objects are both mapped into a Hilbert space within which the matrix entries give the scalar products from which the classification or regression function is constructed. If the dyadic data was produced by a measurement device, then this assumption is based on consideration in Section 2.5 and Appendix A which state that measurements are projections of object feature vectors $\boldsymbol{x}_\phi$ onto a limited set of P

[3]In this case, sphering must be based on the computationally expensive kernel PCA and regularization as in sections 2.4.3 and 2.4.5 is not possible.

complex features $\boldsymbol{z}_\omega$, i.e. the $\boldsymbol{z}_\omega$ are the only measurable and acessible directions in feature space. The value K_{ij} of a complex feature $\boldsymbol{z}_\omega^j$ for an object $\boldsymbol{x}_\phi^i$ is then given by the dot product

$$K_{ij} \;=\; \left\langle \boldsymbol{x}_\phi^i, \boldsymbol{z}_\omega^j \right\rangle \quad . \tag{7}$$

In analogy to the index "ϕ" for $\boldsymbol{x}_\phi$, the index "ω" indicates that we will later assume that the vectors $\boldsymbol{z}_\omega^i$ are images in $\mathbb{R}^N$ of a map ω which is induced by either a kernel or a measurement function. A mathematical foundation of the ansatz eq. (7) is given in Appendix A.

Let $\boldsymbol{Z}_\omega := \left(\boldsymbol{z}_\omega^1, \boldsymbol{z}_\omega^2, \ldots, \boldsymbol{z}_\omega^P\right)$ be the matrix of the complex features. Then we can summarize our (incomplete) knowledge about the set of objects X_ϕ using the data matrix $\boldsymbol{K}$, where

$$\boldsymbol{K} \;=\; \boldsymbol{X}_\phi^\top \; \boldsymbol{Z}_\omega \quad . \tag{8}$$

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 a document data set we could identify $\boldsymbol{K}$ with the matrix of word frequencies. Hence we assume, that $\boldsymbol{x}_\phi$ and $\boldsymbol{z}_\omega$ live in a space of hidden causes which are responsible for the different attributes of the objects. The complex features $\left\{\boldsymbol{z}_\omega^j\right\}$ span a subspace of the original feature space, but we do not require them to be orthogonal, normalized, or linearly independent. If we set $\boldsymbol{z}_\omega^j \;=\; \boldsymbol{e}^j$ (jth Cartesian unit vector), that is $\boldsymbol{Z}_\omega \;=\; \boldsymbol{I}$, $K_{ij} \;=\; x_j^i$ and $P \;=\; N$, the "new" description, eq. (8), is fully equivalent to the "old" description using the original feature vectors $\boldsymbol{x}_\phi$.

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

$$c(y_i, f(\boldsymbol{x}_\phi^i)) \;=\; \frac{1}{2} r_i^2 \;, \tag{9}$$

where

$$r_i \;=\; f(\boldsymbol{x}_\phi^i) - \; y_i \;=\; \left\langle \boldsymbol{w}, \boldsymbol{x}_\phi^i \right\rangle \;+\; b \;-\; y_i \tag{10}$$

is the residual error for a data point $\boldsymbol{x}_\phi^i$. The total residual error on the training set, the mean squared error, is

$$R_{\mathrm{emp}}\left[f_{\boldsymbol{w},b}\right] \;=\; \frac{1}{L} \sum_{i=1}^{L} c\left(y_i, f(\boldsymbol{x}_\phi^i)\right) \quad . \tag{11}$$

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

$$\nabla_{\boldsymbol{w}} R_{\mathrm{emp}}\left[f_{\boldsymbol{w},b}\right] \;=\; \frac{1}{L} \; \boldsymbol{X}_\phi \; \left(\boldsymbol{X}_\phi^\top \boldsymbol{w} \;+\; b\boldsymbol{1} \;-\; \boldsymbol{y}\right) \;=\; \boldsymbol{0} \tag{12}$$

and

$$\frac{\partial R_{\mathrm{emp}}[f]}{\partial b} \;=\; \frac{1}{L} \sum_i r_i \;=\; b \;+\; \frac{1}{L} \sum_i \left(\left\langle \boldsymbol{w}, \boldsymbol{x}_\phi^i \right\rangle \;-\; y_i\right) \;=\; 0 \;, \tag{13}$$

where the labels for all objects in the training set are summarized by a label vector $\boldsymbol{y}$. Since the quadratic loss function is convex in $\boldsymbol{w}$ and b, only one minimum exists if $\boldsymbol{X}_\phi\ \boldsymbol{X}_\phi^\top$ has full rank. If $\boldsymbol{X}_\phi\ \boldsymbol{X}_\phi^\top$ is singular, then all points of minimal value correspond to a subspace of $\mathbb{R}^N$. From eq. (13) we obtain

$$b \;=\; -\frac{1}{L}\ \sum_{i=1}^{L} \left(\langle \boldsymbol{w}, \boldsymbol{x}_\phi^i \rangle \;-\; y_i \right) \;=\; -\frac{1}{L}\ \left(\boldsymbol{w}^\top \boldsymbol{X}_\phi - \boldsymbol{y}^\top \right) \mathbf{1}\ . \tag{14}$$

Condition eq. (12) implies, that the directional derivative should be zero along any direction in feature space, including the directions of the complex feature vectors $\boldsymbol{z}_\omega$. We, therefore, obtain

$$\begin{aligned} &\frac{dR_{\mathrm{emp}}\left[f_{\boldsymbol{w}\ +\ t\ \boldsymbol{z}_\omega^j, b} \right]}{dt} \;=\; \left(\boldsymbol{z}_\omega^j \right)^\top\ \nabla_{\boldsymbol{w}} R_{\mathrm{emp}}\left[f_{\boldsymbol{w},b} \right] \\ &=\ \frac{1}{L}\ \left(\boldsymbol{z}_\omega^j \right)^\top\ \boldsymbol{X}_\phi\ \left(\boldsymbol{X}_\phi^\top \boldsymbol{w}\ +\ b\mathbf{1}\ -\ \boldsymbol{y} \right)\ =\ 0\ , \end{aligned} \tag{15}$$

and, combining all complex features,

$$\begin{aligned} &\frac{1}{L}\ \boldsymbol{Z}_\omega^\top\ \boldsymbol{X}_\phi\ \left(\boldsymbol{X}_\phi^\top \boldsymbol{w}\ +\ b\mathbf{1}\ -\ \boldsymbol{y} \right)\ =\ \frac{1}{L}\ \boldsymbol{K}^\top \left(\boldsymbol{X}_\phi^\top \boldsymbol{w}\ +\ b\mathbf{1}\ -\ \boldsymbol{y} \right) \\ &=\ \frac{1}{L}\ \boldsymbol{K}^\top \boldsymbol{r}\ =\ \mathbf{0}\ . \end{aligned} \tag{16}$$

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

$$\begin{aligned} \sigma_j \;&=\; \frac{1}{L} \sum_{i=1}^{L}\ \left\langle \boldsymbol{x}_\phi^i, \boldsymbol{z}_\omega^j \right\rangle\ r_i\ =\ \frac{1}{L}\ \left[\boldsymbol{K}^\top \boldsymbol{r} \right]_j \\ &=\ \frac{dR_{\mathrm{emp}}\left[f_{\boldsymbol{w}\ +\ t\ \boldsymbol{z}_\omega^j, b} \right]}{dt}\ =\ 0\ . \end{aligned} \tag{17}$$

2.4 The Potential Support Vector Machine (P-SVM)

2.4.1 The Basic P-SVM

The new objective from eq. (6) and the new constraints from eq. (16) constitute a new procedure of selecting a classifier or a regression function. The number of constraints 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. Because the mean squared error of a linear function $f_{\boldsymbol{w},b}$ is a convex function of the parameters $\boldsymbol{w}$ and b, the constraints can always be fulfilled at its minimum.[4]. Therefore, f is chosen from all 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.

[4] $\boldsymbol{w}\ =\ \left(\boldsymbol{X}_\phi^\top \right)^*\ (\boldsymbol{y}\ -\ b\ \mathbf{1})$ fulfills the constraints, where $\boldsymbol{A}^*$ is the pseudo-inverse of $\boldsymbol{A}$.

If $\boldsymbol{K}$ has at least rank L (number of training examples), then $\boldsymbol{r} = \mathbf{0}$ is always enforced (cf. eq. (16)). Consequently, overfitting occurs and a regularization scheme is needed.

Before a regularization scheme can be defined, however, 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}. Since we are interested in the latter the most appropriate measure would be Pearson's correlation coefficient

$$\hat{\sigma}_j \;=\; \frac{\sum_{i=1}^{L} \left(K_{ij} \;-\; \bar{K}_j\right) \; (r_i \;-\; \bar{r})}{\sqrt{\sum_{i=1}^{L} \left(K_{ij} \;-\; \bar{K}_j\right)^2} \sqrt{\sum_{i=1}^{L} \; (r_i \;-\; \bar{r})^2}} \;\; , \tag{18}$$

where $\bar{r} \;=\; \frac{1}{L} \; \sum_{i=1}^{L} \; r_i$ 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,

$$\frac{1}{L} \sum_{i=1}^{L} \; \left(K_{ij} \;-\; \bar{K}_j\right)^2 \;=\; 1 \quad \text{and} \quad \bar{K}_j \;=\; \frac{1}{L} \; \sum_{i=1}^{L} \; K_{ij} \;=\; 0 \; , \tag{19}$$

we obtain

$$\sigma_j \;=\; \frac{1}{L} \sum_{i=1}^{L} \; K_{ij} \; r_i \;\; = \; \hat{\sigma}_j \; \frac{1}{\sqrt{L}} \; \|\boldsymbol{r} \;-\; \bar{r}\mathbf{1}\|_2 \;\; . \tag{20}$$

Because $\bar{r} = 0$ (cf. eq. (13)), the mixed moments are now proportional to the correlation coefficient $\hat{\sigma}_j$ with a proportionality constant which is independent of the complex feature $\boldsymbol{z}_\omega^j$ and σ_j can be used instead of $\hat{\sigma}_j$ to formulate the constraints.

If the data vectors are normalized, the term $\boldsymbol{K}^\top \mathbf{1}$ vanishes and we obtain the basic P-SVM optimization problem

Basic P-SVM optimization problem

$$\begin{array}{ll} \min\limits_{\boldsymbol{w}} & \frac{1}{2} \; \|\boldsymbol{X}_\phi^\top \; \boldsymbol{w}\|^2 \\ \text{s.t.} & \boldsymbol{K}^\top \; \left(\boldsymbol{X}_\phi^\top \; \boldsymbol{w} \;-\; \boldsymbol{y}\right) \;=\; \mathbf{0} \; , \end{array} \tag{21}$$

The offset b of the classification or regression function is given by eq. (14) which to (see (Hochreiter and Obermayer, 2004a), Appendix A)

$$b \;=\; \frac{1}{L} \; \sum_{i=1}^{L} y_i \; . \tag{22}$$

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.

2.4.2 The Kernel Trick

Following the standard "support vector" way to derive learning rules for perceptrons, we have so far considered linear classifiers only and an appropriate feature space $\mathcal{X}_\phi$ within which the classification problem is linear separable. Most practical classification problems, however, require non-linear classification boundaries which makes the construction of a proper feature space necessary. In analogy to the standard SVM, we now invoke the kernel trick.

Let $\boldsymbol{x}^i$ and $\boldsymbol{z}^j$ be feature vectors, which describe the "column" and the "row" objects of the dataset. We then choose a kernel function $k\left(\boldsymbol{x}^i, \boldsymbol{z}^j\right)$ and compute the matrix $\boldsymbol{K}$ of relations between "column" and "row" objects:

$$K_{i,j} \;=\; k\left(\boldsymbol{x}^i, \boldsymbol{z}^j\right) \;=\; \left\langle \phi\left(\boldsymbol{x}^i\right), \omega\left(\boldsymbol{z}^j\right)\right\rangle \;=\; \left\langle \boldsymbol{x}^i_\phi, \boldsymbol{z}^j_\omega \right\rangle \;, \tag{23}$$

where $\boldsymbol{x}^i_\phi \;=\; \phi\left(\boldsymbol{x}^i\right)$ and $\boldsymbol{z}^j_\omega \;=\; \omega\left(\boldsymbol{z}^j\right)$. In Appendix A it is shown that any L^2-kernel corresponds (for almost all points) to a dot product in a Hilbert space in the sense of eq. (23) and corresponds to an (implicit) mapping into a feature space within which a linear classifier is constructed. In the following chapters we will, therefore, distinguish between the actual measurements $\boldsymbol{x}^i$ and $\boldsymbol{z}^j$ and the feature vectors $\boldsymbol{x}^i_\phi$, and $\boldsymbol{z}^j_\omega$ "induced" by the kernel k. Potential choices for "row" objects and their vectorial description are (1) $\boldsymbol{z}^j \;=\; \boldsymbol{x}^j, P = L$, (standard construction of a Gram matrix), (2) $\boldsymbol{z}^j \;=\; \boldsymbol{e}^j, P = N$ ("elementary" features), (3) $\boldsymbol{z}^j$ is the jth cluster center of a clustering algorithm applied to the vectors $\boldsymbol{x}^i$ (example for a "complex" feature), or (4) $\boldsymbol{z}^j$ is the jth vector of an PCA or ICA preprocessing (another example for a "complex" feature).

If the entries $K_{i,j}$ of the data matrix are directly measured, the application of the kernel trick needs additional considerations. In appendix A we show, that - if the measurement process can be expressed through a kernel $k\left(x^i, z^j\right)$, which takes a column object x^i and a row object z^j and outputs a number - the matrix $\boldsymbol{K}$ of relations between the "row" and "column" objects can be interpreted as a dot product in some features space:

$$K_{i,j} \;=\; \left\langle \boldsymbol{x}^i_\phi, \boldsymbol{z}^j_\omega \right\rangle \;, \tag{24}$$

where $\boldsymbol{x}^i_\phi \;=\; \phi\left(x^i\right)$ and $\boldsymbol{z}^j_\omega \;=\; \omega\left(z^j\right)$. Note, that we distinguish between an object x^i and its associated feature vectors $\boldsymbol{x}^i$ or $\boldsymbol{x}^i_\phi$, leading to differences in the definition of k for the cases of vectorial and (measured) dyadic data. Eq. (24) justifies the P-SVM approach, which was derived for the case of linear predictors, also for measured data.

2.4.3 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 the correct choice of $\boldsymbol{w}$ would lead to a large increase

of the objective function otherwise. We obtain

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

for the primal problem.

Above regularization scheme makes the optimization problem robust against "outliers". A large value of a slack variable indicates, that the particular "row" object 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 evaluate the Lagrangian L,

$$\begin{aligned} L \;=\; & \frac{1}{2}\,\boldsymbol{w}^\top\,\boldsymbol{X}_\phi\,\boldsymbol{X}_\phi^\top\,\boldsymbol{w}\;+\;C\mathbf{1}^\top\left(\boldsymbol{\xi}^+\;+\;\boldsymbol{\xi}^-\right) \\ & -\;\left(\boldsymbol{\alpha}^+\right)^\top\;\left(\boldsymbol{K}^\top\left(\boldsymbol{X}_\phi^\top\;\boldsymbol{w}\;-\;\boldsymbol{y}\right)\;+\;\boldsymbol{\xi}^+\right) \\ & +\;\left(\boldsymbol{\alpha}^-\right)^\top\;\left(\boldsymbol{K}^\top\left(\boldsymbol{X}_\phi^\top\;\boldsymbol{w}\;-\;\boldsymbol{y}\right)\;-\;\boldsymbol{\xi}^-\right) \\ & -\;\left(\boldsymbol{\mu}^+\right)^\top\;\boldsymbol{\xi}^+\;-\;\left(\boldsymbol{\mu}^-\right)^\top\;\boldsymbol{\xi}^-\;\;, \end{aligned} \tag{26}$$

$$\tag{27}$$

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. (25). The optimality conditions (Schölkopf and Smola, 2002) require that

$$\begin{aligned} \nabla_{\boldsymbol{w}} L \;&=\; \boldsymbol{X}_\phi\,\boldsymbol{X}_\phi^\top\,\boldsymbol{w}\;-\;\boldsymbol{X}_\phi\,\boldsymbol{K}\,\boldsymbol{\alpha} \\ &=\; \boldsymbol{X}_\phi\,\boldsymbol{X}_\phi^\top\,\boldsymbol{w}\;-\;\boldsymbol{X}_\phi\,\boldsymbol{X}_\phi^\top\boldsymbol{Z}_\omega\,\boldsymbol{\alpha}\;=\;\mathbf{0}\;, \end{aligned} \tag{28}$$

where we used the abbreviation $\boldsymbol{\alpha} = \boldsymbol{\alpha}^+ \;-\; \boldsymbol{\alpha}^-$ ($\alpha_i = \alpha_i^+ \;-\; \alpha_i^-$). In order to ensure eq. (28) and its equivalent equation $\boldsymbol{X}_\phi\,\boldsymbol{X}_\phi^\top\,\boldsymbol{w} \;=\; \boldsymbol{X}_\phi\,\boldsymbol{X}_\phi^\top\boldsymbol{Z}_\omega\,\boldsymbol{\alpha}$, we set

$$\boldsymbol{w} \;=\; \boldsymbol{Z}_\omega\,\boldsymbol{\alpha}\;\;. \tag{29}$$

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

P-SVM classification optimization problem

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha}} & \frac{1}{2}\boldsymbol{\alpha}^\top \boldsymbol{K}^\top \boldsymbol{K}\, \boldsymbol{\alpha} \;-\; \boldsymbol{y}^\top \boldsymbol{K}\, \boldsymbol{\alpha} \\ \text{s.t.} & -C\,\mathbf{1} \;\leq\; \boldsymbol{\alpha} \;\leq\; C\mathbf{1}\;, \end{array} \tag{30}$$

which now only depends on the data via the kernel or data matrix $\boldsymbol{K}$. The dual problem is solved by a Sequential Minimal Optimization (SMO) technique which is described in (Hochreiter and Obermayer, 2004a), which is essential if many complex features are used, because the $P \times P$ matrix $\boldsymbol{K}^\top \boldsymbol{K}$ enters the dual formulation.

Finally, the classification function f has to be constructed using the optimal values of the Lagrange parameters $\boldsymbol{\alpha}$.

P-SVM classification function

$$f(\boldsymbol{x}_\phi) \;=\; \sum_{j=1}^{P} \alpha_j \; K_{(x)j} + \; b\;,$$

where the expansion eq. (29) has been used for the weight vector $\boldsymbol{w}$ and b is given by eq. (22).

The classifier based on eq. (31) depends on the weighting coefficients α_j, which were determined during optimization, on b, which can be computed directly, and on the measured values $K_{(x)j}$ for the new object x. 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

$$\frac{dR_{\text{emp}}\left[f_{\boldsymbol{w}\,+\,t\;\boldsymbol{z}_\omega^j\;,\;b}\right]}{dt} \;=\; \sigma_j \;=\; \frac{\partial R_{\text{emp}}[f]}{\partial \alpha_j}\;. \tag{31}$$

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 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 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 classification using a toy example. The data set consists of 70 objects x, 28 from class 1 and 42 from class 2, which are described by 2D-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 $k\left(\boldsymbol{x}^i, \boldsymbol{x}^j\right) = \sin\left(\theta \left\|\boldsymbol{x}^i - \boldsymbol{x}^j\right\|^2\right)$ leading to an indefinite Gram matrix. 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 because it is better adjusted to the "oscillatory" regions of class membership, leading to a smaller number of support vectors and to a smaller test error. In general, the value of θ has to be selected using standard model selection techniques. A large value of θ leads to a more "complex" set of classifiers, reduces the classification error on the training set, but increases the error on the test set. Non-Mercer kernels extend the range of kernels which are currently used and, therefore, opens up a new direction of research for kernel design.

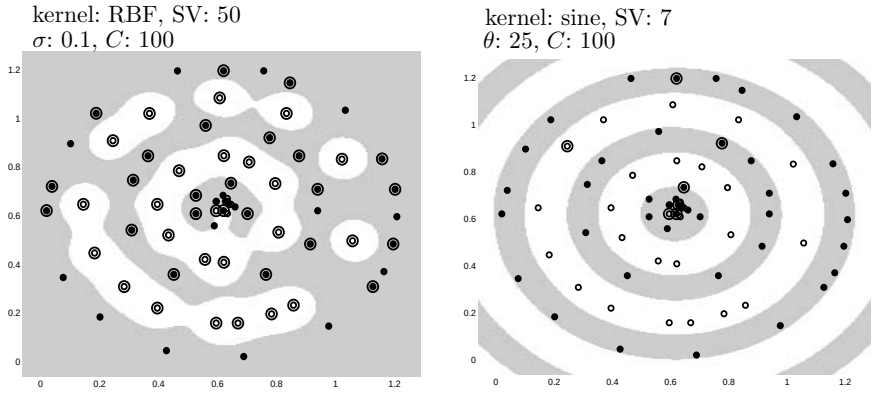

Figure 4: Application of the P-SVM method to a toy classification problem. Objects are described by two-dimensional feature vectors $\boldsymbol{x}$, and 70 feature vectors were generated of which 28 belong to class 1 (open circles) and 42 belong to class 2 (solid circles). A Gram matrix was constructed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

2.4.4 The P-SVM for Regression

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 task. In regression the term $\left\|\boldsymbol{X}_\phi^\top \boldsymbol{w}\right\|^2 = \left\|\boldsymbol{X}_\phi^\top \hat{\boldsymbol{w}}\right\|^2 \|\boldsymbol{w}\|^2$, $\hat{\boldsymbol{w}} := \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|}$, is the product of a term which expresses the deviation of the data from the zero-isosurface of 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 $\boldsymbol{x}_\phi$ to have zero mean (see eq. (19)) and by normalizing the attributes y_i such that $b = 0$ (see eq. (22)), then $\boldsymbol{X}_\phi^\top \hat{\boldsymbol{w}}$ 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 $\boldsymbol{w}$. If $f(\boldsymbol{x}_\phi^i) = \left\langle \boldsymbol{w}, \boldsymbol{x}_\phi^i \right\rangle + b$ and the length of the vectors $\boldsymbol{x}_\phi$ is bounded by B, then the deviation of f from offset b is bounded by:

$$\left\| f(\boldsymbol{x}_\phi^i) - b \right\| = \left\| \langle \boldsymbol{w}, \boldsymbol{x}_\phi^i \rangle \right\| \leq \|\boldsymbol{w}\| \left\| \boldsymbol{x}_\phi^i \right\| \leq \|\boldsymbol{w}\| B \,, \tag{32}$$

where the first "$\leq$" follows from the Cauchy-Schwarz inequality, hence a smaller value of $\|\boldsymbol{w}\|$ leads to a smoother regression function. This tradeoff between smoothness and residual error is also reflected by eq. (64) in Appendix A which shows that eq. (6) is the L^2-norm of the function f. The discussion in Section 2.3 and beginning of Section 2.4 also showed, that the constraints of vanishing mixed moments carry over to regression problems with the only modification, that the target values y_i in eqs. (25) are real rather than binary 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. (25), and its corresponding dual, eqs. (30), also for the regression setting.

Fig. 5 shows the application of the P-SVM to a toy regression example (pairwise data). 50 data points are randomly chosen from the true function (dashed line) and i.i.d. Gaussian noise with mean 0 and standard deviation 0.2 is 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. The hyperparameter C controls the sensitivity against outliers: A smaller value of C increases the error at $x = 0$ but also the number of support vectors. The width σ of the RBF-kernel controls the overall smoothness of the regressor: A larger value of σ increases the error at $x = 0$ without increasing the number of support vectors (cf. arrows in bold in Fig. 5).

2.4.5 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, 2004b). 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 number of "informative" features from the set of "row" objects. The set of "support features" may then be used as input to an arbitrary predictor, e.g. another P-SVM, a standard SVM or a K-nearest-neighbor classifier.

Noisy measurements can lead to spurious mixed moments, i.e. complex features may 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

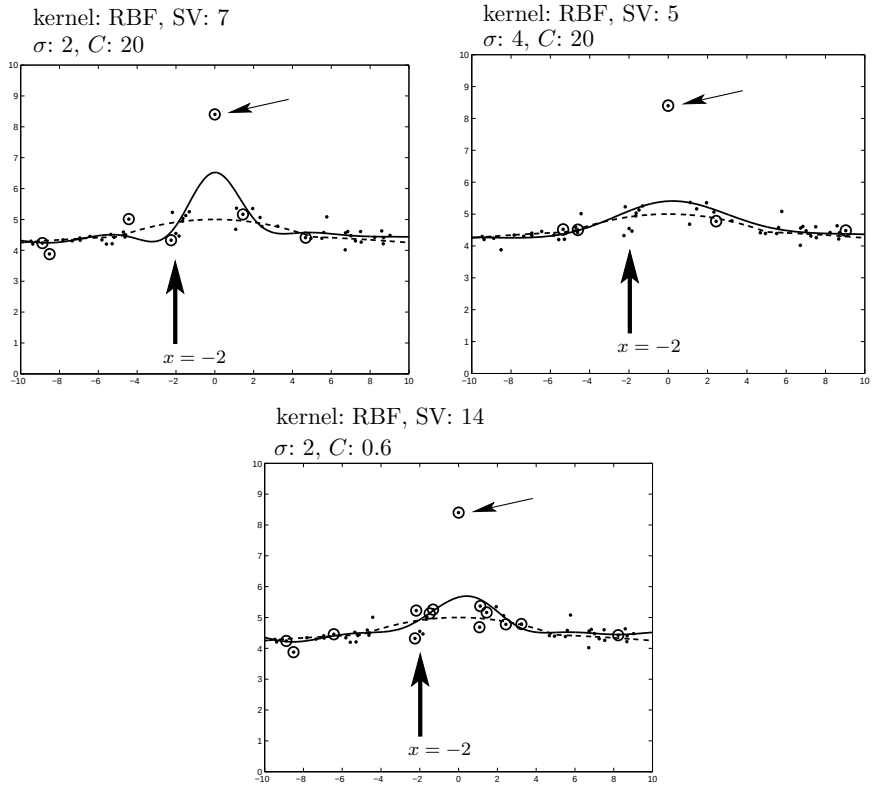

Figure 5: Application of the P-SVM method to a toy regression problem. Objects (small dots), described by the x-coordinate, were generated by randomly choosing points from the true function (dashed line) and adding Gaussian noise with mean 0 and standard deviation 0.2 to the y-component of each data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

and modify the constraints in problem eqs. (21) according to

$$\left\|\boldsymbol{K}^\top \left(\boldsymbol{X}_\phi^\top \boldsymbol{w} - \boldsymbol{y}\right)\right\|_\infty \leq \epsilon \,, \tag{33}$$

which can be written as

$$\boldsymbol{K}^\top \left(\boldsymbol{X}_\phi^\top \boldsymbol{w} - \boldsymbol{y}\right) - \epsilon\, \mathbf{1} \leq \mathbf{0} \,, \quad \boldsymbol{K}^\top \left(\boldsymbol{X}_\phi^\top \boldsymbol{w} - \boldsymbol{y}\right) + \epsilon\, \mathbf{1} \geq \mathbf{0} \tag{34}$$

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.

The value of ϵ directly correlates with the strength of the measurement noise, and can be determined a priori if it is known. If this is not the case, ϵ serves as a hyperparameter and its value can be determined using model selection techniques. Note, that data vectors have to be normalized (cf. eqs. (19)) before applying the P-SVM, because otherwise a global value of ϵ would not suffice.

Combining eq. (6) and eqs. (34) we then obtain the primal optimization problem

$$\begin{array}{ll} \min\limits_{\boldsymbol{w}} & \frac{1}{2}\, \|\boldsymbol{X}_\phi^\top\, \boldsymbol{w}\|^2 \\ \text{s.t.} & \boldsymbol{K}^\top\, \left(\boldsymbol{X}_\phi^\top\, \boldsymbol{w}\, -\, \boldsymbol{y}\right)\, +\, \epsilon\, \mathbf{1}\, \geq\, \mathbf{0} \\ & \boldsymbol{K}^\top\, \left(\boldsymbol{X}_\phi^\top\, \boldsymbol{w}\, -\, \boldsymbol{y}\right)\, -\, \epsilon\, \mathbf{1}\, \leq\, \mathbf{0} \end{array} \tag{35}$$

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

$$\begin{aligned} L \;=\; & \frac{1}{2}\, \boldsymbol{w}^\top\, \boldsymbol{X}_\phi\, \boldsymbol{X}_\phi^\top\, \boldsymbol{w} \\ & -\, \left(\boldsymbol{\alpha}^+\right)^\top\, \left(\boldsymbol{K}^\top \left(\boldsymbol{X}_\phi^\top\, \boldsymbol{w}\, -\, \boldsymbol{y}\right)\, +\, \epsilon\, \mathbf{1}\right) \\ & +\, \left(\boldsymbol{\alpha}^-\right)^\top\, \left(\boldsymbol{K}^\top \left(\boldsymbol{X}_\phi^\top\, \boldsymbol{w}\, -\, \boldsymbol{y}\right)\, -\, \epsilon\, \mathbf{1}\right)\quad , \end{aligned} \tag{36}$$

where we have used the notation from Section 2.4.3. The vector $\boldsymbol{w}$ is again expressed through the complex features,

$$\boldsymbol{w}\; =\; \boldsymbol{Z}_\omega\, \boldsymbol{\alpha}\;, \tag{37}$$

and we obtain the dual formulation of eq. (35):

P-SVM feature selection optimization problem

$$\begin{array}{ll} \min\limits_{\boldsymbol{\alpha}^+, \boldsymbol{\alpha}^-} & \frac{1}{2}\left(\boldsymbol{\alpha}^+\, -\, \boldsymbol{\alpha}^-\right)^\top\, \boldsymbol{K}^\top\, \boldsymbol{K}\, \left(\boldsymbol{\alpha}^+\, -\, \boldsymbol{\alpha}^-\right) \\ & -\, \boldsymbol{y}^\top\, \boldsymbol{K}\, \left(\boldsymbol{\alpha}^+\, -\, \boldsymbol{\alpha}^-\right)\, +\, \epsilon\, \mathbf{1}^\top \left(\boldsymbol{\alpha}^+\, +\, \boldsymbol{\alpha}^-\right) \\ \text{s.t.} & \mathbf{0}\, \leq\, \boldsymbol{\alpha}^+\;,\;\; \mathbf{0}\, \leq\, \boldsymbol{\alpha}^-\;\; . \end{array} \tag{38}$$

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 fast Sequential Minimal Optimization (SMO) technique (see (Hochreiter and Obermayer, 2004a)).

Finally, let us address the relationship between the value of a Lagrange multiplier α_j and the "importance" of the corresponding complex feature $\boldsymbol{z}_\omega^j$ 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 $\boldsymbol{z}_\omega^j$ is given by

$$\begin{aligned} & R_{\text{emp}}\left[f_{\boldsymbol{w} \,+\, \beta\, \boldsymbol{z}_\omega^j \,,\, b}\right] \;-\; R_{\text{emp}}\left[f_{\boldsymbol{w},b}\right] \\ &= \;\beta\;\sigma_j \;+\; \frac{\beta^2}{2\;L}\sum_i K_{ij}^2 \;=\; \beta\;\sigma_j \;+\; \frac{\beta^2}{2} \\ &\leq \;\frac{\epsilon\;|\beta|}{L} \;+\; \frac{\beta^2}{2}\;, \end{aligned} \tag{39}$$

because the constraints eq. (34) ensure that $|\sigma_j|\;L \;\leq\; \epsilon$. If a complex feature $\boldsymbol{z}_\omega^j$ is completely removed, then $\beta \;=\; -\;\alpha_j$ and

$$R_{\text{emp}}\left[f_{\boldsymbol{w} \,-\, \alpha_j\; \boldsymbol{z}_\omega^j \,,\, b}\right] \;-\; R_{\text{emp}}\left[f_{\boldsymbol{w},b}\right] \;\leq\; \tfrac{\epsilon\;|\alpha_j|}{L} \;+\; \tfrac{\alpha_j^2}{2}\;. \tag{40}$$

The Lagrange parameter α_j is directly related to the increase in the empirical error when a feature is removed. Therefore, $\boldsymbol{\alpha}$ serve as importance measures for the complex features and allows to rank features according to the absolute value of its components.

In the following, we illustrate the application of the P-SVM approach to feature selection using a toy example, which considers a classification task. The data set consists of 50 "column" objects x, 25 from each class, which are described by 2D-feature vectors $\boldsymbol{x}$ (open and solid circles in Fig. 6). 50 "row" objects z were randomly selected by choosing their 2D-feature vectors $\boldsymbol{z}$ 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 $k\left(\boldsymbol{x}^i, \boldsymbol{z}^j\right) \;=\; \exp\left(-\frac{1}{2\;\sigma^2}\;\|\boldsymbol{x}^i \;-\; \boldsymbol{z}^j\|^2\right)$ with std $\sigma = 0.2$. 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 ("column" objects) is described (and detected) by one or two feature objects.

The number of selected features depends on ϵ, and on σ, which determines how the "strength" of a complex feature decreases with the distances $\|\boldsymbol{x}^i \;-\; \boldsymbol{z}^j\|$. Smaller ϵ or larger σ would result in more complex features assigned to every data group.

2.5 The Dot Product Interpretation of Dyadic Data

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.

As shown in Appendix A, above question reduces to the question whether or not the following three conditions hold:

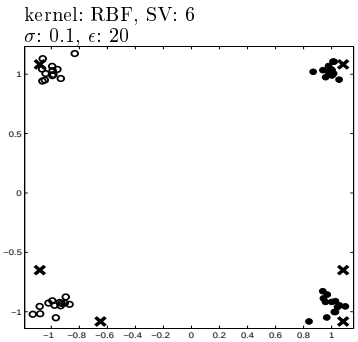

Figure 6: Application of the P-SVM method to a toy feature selection problem for a classification task. "Column" and "row" objects are described by two-dimensional feature vectors $\boldsymbol{x}$ and $\boldsymbol{z}$, respectively. Feature vectors for 50 "column" objects, 25 from each class (open and solid circles), were generated randomly by choosing one of the centers from $\{(1,1),(1,-1),(-1,1),(-1,-1)\}$ with equal probability and then constructing a 2D-feature vecotro by adding to each coordinate a random number drawn from a Gaussian with mean 0 and standard deviation 0.1. Feature vectors of 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 support features selected by the P-SVM method.

- (1) "Column" objects ("samples") x are from a set $\mathcal{X}$ which can be completed to a measure space.
- (2) "Row" objects 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 $k\,(x,z)$ which is from $L^2(\mathcal{X},\mathcal{Z})$.

Conditions (1) and (2) can easily fulfilled by defining a suitable σ-algebra on the sets; condition (3) holds if k is bounded and the sets $\mathcal{X}$ and $\mathcal{Z}$ are compact. Condition (3) equates the evaluation of a kernel as known from standard SVMs with physical measurements, and physical characteristics of the measurement device determines the properties of the kernel, e.g. boundedness and continuity. But can a measurement process indeed be expressed through a kernel?. 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.

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 with C- and the ν-SVMs for different kernels. Secondly, we apply the P-SVM to a measured (rather than constructed) pairwise ("protein", see Hochreiter and Obermayer, 2004a for others) and dyadic 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 microarray data (reported from Hochreiter and Obermayer, 2004b) and for the "protein" and "World Wide Web" data sets from part one.

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 the data sets "german" and "banana" we restricted the training set to the first 200 examples of the original training set in order to keep hyperparameter selection feasible.

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 kernel Fisher discriminant analysis (KFD, Mika et al., 1999), C-, ν-, and P-SVM. Because KFD only selects a direction in input space onto which all data points are projected, we must select a classifier for the resulting one-dimensional classification task. We follow (Schölkopf and Smola, 2002) and classify a data point according to its posterior probability under the assumption of a Gaussian distribution for each label. 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, except that the ν values of the ν-SVM were selected from $\{0.1, \ldots, 0.9\}$ in contrast to the selection of C for which a logarithmic scale was used. To test the significance of the differences

in generalization performance (percent of misclassifications), we first performed a test for the "difference of two proportions" for each training/test set pair (Dietterich, 1998). The "difference of two proportions" is the difference of the misclassification rates of two models on the test set, where the models are selected on the training set by the two methods which are to be compared. After this test we adjusted the false discovery rate through the "Benjamini Hochberg Procedure" (Benjamini and Hochberg, 1995) which was recently shown to be correct for dependent outcomes of the tests (Benjamini and Yekutieli, 2001). The fact that the tests can be dependent is important because training and test sets overlap for the different training/test set pairs. The false detection rate was controlled at 5 %. We counted for each pair of methods the selected models from the first method which perform significantly (5 % level) better than the selected models from the second method.

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 the somewhat different model selection results for the hyperparameters C and ν. Best and second best results are indicated by bold and italic numbers. The significance tests did not reveal significant differences in generalization performance for most of the cases (for details see `http://ni.cs.tu-berlin.de/publications/psvm_sup`). The ν-SVM with the Plummer kernel however performed slightly (less than 5 significant differences out of 100) worse than the other methods[5]. For the "banana" data set, P-SVM with "RBF" was significantly the best, followed by P-SVM with "PLU", ν-SVM with "RBF", and ν-SVM with "PLU" (other methods perform significantly worse).

The UCI-benchmark result shows that the P-SVM is competitive to other state-of-the-art methods for a standard problem setting (measurement matrix equivalent to the Gram matrix). Although the P-SVM method never performed significantly worse, it generally required fewer support vectors than other SVM approaches. This was also true for the cases, where the P-SVM's performance was significantly better.

3.1.2 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 2 summarizes the classification results, which were obtained with the generalized SVM (G-SVM, Graepel et al., 1999; Mangasarian, 1998) and the P-SVM method. Since the G-SVM interprets the columns of the data matrix as feature vectors for the column objects and applies a standard ν-SVM to these

[5]The ν-SVM with POL performed better on the data set "thyroid" than the P-SVM with PLU and KFD with POL but not better than others.

	RBF	POL	PLU
Thyroid			
C-SVM	5.11 (2.34)	*4.51* (2.02)	4.96 (2.35)
ν-SVM	5.15 (2.23)	**4.04** (2.12)	4.83 (2.03)
KFD	4.96 (2.24)	6.52 (3.18)	5.00 (2.26)
P-SVM	4.71 (2.06)	9.44 (3.12)	5.08 (2.18)
Heart			
C-SVM	16.67 (3.51)	18.26 (3.50)	*16.33* (3.47)
ν-SVM	16.87 (3.87)	17.44 (3.90)	18.47 (7.81)
KFD	17.82 (3.45)	22.53 (3.37)	17.80 (3.86)
P-SVM	**16.18** (3.66)	16.67 (3.40)	16.54 (3.64)
Breast-Cancer			
C-SVM	26.94 (5.18)	26.87 (4.79)	*26.48* (4.87)
ν-SVM	27.69 (5.62)	26.69 (4.73)	30.16 (7.83)
KFD	27.53 (4.19)	31.30 (6.11)	27.19 (4.72)
P-SVM	26.78 (4.58)	**26.40** (4.54)	26.66 (4.59)
Banana			
C-SVM	11.88 (1.11)	12.09 (0.96)	11.81 (1.07)
ν-SVM	11.67 (0.90)	12.72 (2.16)	11.74 (0.98)
KFD	12.32 (1.12)	14.04 (3.89)	12.14 (0.96)
P-SVM	*11.59* (0.96)	14.93 (2.09)	**11.52** (0.93)
German			
C-SVM	26.51 (2.60)	27.27 (3.23)	26.88 (3.12)
ν-SVM	27.14 (2.84)	27.13 (3.06)	28.60 (6.27)
KFD	26.58 (2.95)	30.96 (3.23)	26.90 (3.15)
P-SVM	*26.45* (3.20)	**25.87** (2.45)	26.65 (2.95)

Table 1: Average percentage of misclassification for the UCI and the "banana" data sets. The table compares results obtained with the kernel Fisher discriminant analysis (KFD), 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), $\left(\langle \boldsymbol{x}^i, \boldsymbol{x}^j \rangle + \eta\right)^\delta$, and Plummer (PLU), $\frac{1}{(\|\boldsymbol{x}^i - \boldsymbol{x}^j\| + \rho)^\zeta}$, 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 numbers in brackets denote standard deviations of the results. C, ν, and kernel parameters were determined using 5–fold cross validation on the training set and usually differed between individual experiments.

vectors (this is also called "feature map" Schölkopf and Smola, 2002), we call the G-SVM simply "ν-SVM" in the following. The table shows the percentage of misclassification for the four two-class classification problems "one class against the rest". The P-SVM yields better classification results although a conservative test for significance was not possible due to the small number of data. However,

The P-SVM selected 180 proteins as support vectors on average, compared to 203 proteins used by the ν-SVM (note that for 10–fold cross validation 203 is the average training size). Here a 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.

protein data					
	Reg.	H-α	H-β	M	GH
Size	—	72	72	39	30
ν-SVM	0.05	1.3	4.0	0.5	0.5
ν-SVM	0.1	1.8	4.5	0.5	0.9
ν-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 2: Percentage of misclassification for the "protein" data set for classifiers obtained with the P-SVM and ν-SVM methods. Column "Reg." lists the value of the regularization parameter (ν for ν-SVM and C for P-SVM). Columns "H-α" to "GH" provide the results for the four classification problems "one class against the rest". The percentage of misclassification was computed using 10–fold cross validation. The best classification results for each problem are shown in bold. Note, that the data matrix contained "measured" values (rather than values computed using a kernel) and was not positive semi-definite.

3.1.3 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 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 the 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 3 summarizes the classification results for the C- 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. Significance tests were performed to evaluate the differences in generalization performance using the 10-fold cross-validated paired t-test (Dietterich, 1998). We considered 48 tasks (4 universities, 4 classes, 3 matrices) and checked for each task whether the C-SVM or the P-SVM performed better using a p-value of 0.05. In 30 tasks the P-SVM had a significantly better performance than the C-SVM, while the C-SVM was never significantly better than the P-SVM (for details see `http://ni.cs.tu-berlin.de/publications/psvm_sup`).

Classification results are better for the asymmetric matrices "in" and "out" than for the symmetric matrix "sym", because there are cases for which highly 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 vectors, online classification is faster than for the C-SVM and – if web pages cease to exist – the P-SVM is more likely not to be affected. Table 4 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.

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 three regression 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

	Course	Faculty	Project	Student
Cornell University				
Size	57	60	52	143
C-SVM (Sym)	11.1 (6.2)	19.7 (5.3)	13.7 (4.0)	50.0 (11.5)
C-SVM (Out)	12.6 (3.1)	15.1 (6.0)	*10.6* (4.9)	22.3 (10.3)
C-SVM (In)	11.1 (4.9)	21.4 (4.3)	14.6 (5.5)	48.9 (15.5)
P-SVM (Sym)	12.3 (3.3)	17.1 (6.2)	15.4 (6.3)	19.1 (5.7)
P-SVM (Out)	*8.6* (3.8)	*14.3* (6.3)	**8.3** (4.9)	**16.9** (7.9)
P-SVM (In)	**7.1** (4.1)	**13.7** (6.6)	10.9 (5.5)	*17.1* (5.7)
Texas University				
Size	52	35	29	129
C-SVM (Sym)	17.2 (9.0)	22.0 (9.1)	19.8 (6.7)	53.5 (11.8)
C-SVM (Out)	*9.5* (5.1)	16.5 (5.8)	20.2 (8.7)	28.9 (11.7)
C-SVM (In)	12.6 (4.9)	20.6 (7.8)	20.9 (5.1)	*16.4* (7.6)
P-SVM (Sym)	15.8 (5.8)	13.6 (7.3)	12.2 (6.2)	25.5 (6.9)
P-SVM (Out)	**8.1** (7.6)	**9.8** (3.6)	**9.8** (3.9)	20.9 (6.7)
P-SVM (In)	12.3 (5.6)	*10.5* (6.3)	**9.4** (4.6)	**13.0** (5.0)
Wisconsin University				
Size	77	36	22	117
C-SVM (Sym)	27.0 (10.0)	22.0 (5.5)	14.0 (6.4)	49.3 (11.1)
C-SVM (Out)	19.3 (7.5)	16.0 (3.8)	10.3 (4.8)	34.3 (10.5)
C-SVM (In)	22.0 (8.6)	16.3 (5.8)	*7.7* (4.5)	24.3 (9.9)
P-SVM (Sym)	18.7 (4.5)	15.0 (9.3)	10.0 (5.4)	34.3 (8.6)
P-SVM (Out)	**12.0** (5.5)	*11.3* (6.5)	*7.7* (4.2)	*23.7* (4.8)
P-SVM (In)	*13.3* (4.4)	**8.7** (8.2)	**6.3** (7.1)	**13.3** (5.9)
Washington University				
Size	169	44	39	151
C-SVM (Sym)	19.6 (6.8)	18.7 (6.8)	10.6 (3.5)	43.6 (8.3)
C-SVM (Out)	**10.6** (4.6)	14.1 (3.0)	14.3 (4.8)	28.2 (9.8)
C-SVM (In)	20.3 (6.4)	20.4 (5.3)	13.8 (4.7)	38.3 (11.9)
P-SVM (Sym)	17.1 (4.4)	13.4 (6.6)	*8.8* (2.1)	20.3 (6.8)
P-SVM (Out)	**10.6** (5.2)	*12.7* (2.9)	**6.7** (3.4)	*17.1* (4.3)
P-SVM (In)	*11.8* (5.6)	**9.2** (6.2)	**6.7** (2.0)	**14.3** (6.9)

Table 3: Percentage of misclassification for the World Wide Web data sets for classifiers obtained with the P-SVM and C-SVM methods. The percentage of misclassification was measured using 10–fold cross-validation. The best results and second best for each data set and classification task are indicated in bold and italics; numbers in brackets denote standard deviations of the results.

around the values for C and σ selected in the first step (65 tests per parameter selection).

	pages	student pages	"in"	"out"	"sym"
Cornell	350	143	17 +12/-21	17 +11/-21	19 +23/-16
Texas	286	129	13 +10/-15	21 +35/-9	26 +33/-20
Wisconsin	300	117	13 +14/-13	24 +32/-19	34 +42/-29
Washington	433	151	14 +18/-12	17 +36/-7	20 +30/-15

Table 4: P-SVM 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.

Table 5 shows the mean squared error and the standard deviation of the results. We also performed a Wilcoxon signed rank test to verify the significance for these results (for details see `http://ni.cs.tu-berlin.de/publications-/psvm_sup`), except for the "robot arm" data set, which has only one training/test set pair, and the "boston housing" data set, which contains too few test examples. On "computer activity" SVR was significantly better (5 % threshold) than BSVR, and on both data sets "computer activity" and "abalone", SVR and BSVR were significantly outperformed by the P-SVM (the P-SVM used fewer support vectors than its competitors).

	robot arm (10^{-3})	boston housing	computer activity	abalone
SVR	**5.84**	10.27 (7.21)	13.80 (0.93)	0.441 (0.021)
BSVR	5.89	12.34 (9.20)	17.59 (0.98)	0.438 (0.023)
P-SVM	5.88	**9.42** (4.96)	**10.28** (0.47)	**0.424** (0.017)

Table 5: Regression results for the UCI data sets. The table shows the mean squared error and its standard deviation in brackets. Best results for each data set are shown in bold. For the "robot arm" data only one data set was available and, therefore, no standard deviation is given.

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 2.4.5).

We first reanalyze the "protein" and "world wide web" data sets of sections 3.1.3 and 3.1.2 and then report results on three DNA microarray data sets. Further feature selection results can be found in (Hochreiter and Obermayer, 2005) where also results for the NIPS feature selection challange are reported and where the P-SVM was the best stand-alone method for selecting a compact feature set, and in (Hochreiter and Obermayer, 2004b), where details of the microarray datasets benchmarks are reported.

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 3.1.2 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 better classification result. Reducing the number of features is beneficial if measurements are costly and if a small increase in prediction error can be tolerated.

Table 6 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 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. The size of the training set is 203. Note, that C was smaller than in the experiments in Section 3.1.2 because large values of ϵ pushed the dual variables $\boldsymbol{\alpha}$ towards zero and, therefore, large values of C 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 % misclassifications compared to about 2 % at the optimum. Since every feature value has to be determined via a sequence alignment, this saving in computation time is essential for large data bases like the Swiss-Prot data base (130,000 proteins), where supplying all pairwise relations is currently impossible.

Table 7 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 3.1.3) 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.

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 6: 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.

"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 7: 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 Micorarray Data Sets

In this subsection we describe the application the P-SVM to real DNA microarray data. All data set consist of tumor tissue samples which were characterized

by the expression values of genes. Samples are labeled according to the outcome of a particular treatment (positive/negative) and the task is to predict the outcome for a new patient. The data was taken from Pomeroy et al. (2002) (brain tumor), Shipp et al. (2002) (lymphoma tumor), and van't Veer et al. (2002) (breast cancer). The P-SVM results are taken from (Hochreiter and Obermayer, 2004b) where the details concerning the data sets and the gene selection procedure based on the P-SVM can be found.

We compare following combinations of feature selection and classification methods:

	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

Table 8 summarizes the results which are taken from the corresponding literature. The P-SVM method outperforms standard methods – in most cases with fewer selected genes.

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 dyadic data, and neither requires square nor positive definite Gram matrices. Therefore, the method can also be used without preprocessing with matrices which are measured and with matrices which are constructed from a vectorial representation using an indefinite 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. The P-SVM approach was compared with several state-of-the-art classification,

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 8: Classification results for DNA microarray data sets, where the leave-one-out error E (% misclassifications) and the number F of features are reported. For breast cancer only the minimal value of E for different thresholds was available, therefore the area under a receiver operating curve is provided in addition.

regression and feature selection methods. Whenever significance tests could be applied, the P-SVM never performed significantly worse than its "competitor", in many cases it performed significantly better. But even if no significant improvement in prediction error could be found, the P-SVM needed less "support features", i.e. less measurements, for evaluating new data objects.

Finally, we have suggested a new interpretation of dyadic data. 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 Anna-Geissler-Stiftung and the Monika-Kuntzner-Stiftung.

A Measurements, Kernels, and Dot Products

In this appendix 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 "dyadic data" set. 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 a finite and almost always exists for an infinite number of "row" objects. The classification or regression function, which is chosen by the P-SVM, exists for all "column" objects.

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 then 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

$$\|f\|_{L^2_\mu} := \left(\int_{\mathcal{U}} f^2 \, d\mu \right)^{\frac{1}{2}} \tag{41}$$

and the set

$$L^2_\mu(\mathcal{U}) := \left\{ f : \mathcal{U} \to \mathbb{R}; f \text{ is } \mu\text{-measurable and } \|f\|_{L^2_\mu} < \infty \right\} . \tag{42}$$

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

$$\langle f, g \rangle_{L^2_\mu(\mathcal{U})} := \int_{\mathcal{U}} f \; g \; d\mu \tag{43}$$

then the Banach space $L^2_\mu(\mathcal{U})$ is a Hilbert space with a dot product $\langle \cdot, \cdot \rangle_{L^2_\mu(\mathcal{U})}$ and scalar body $\mathbb{R}$. 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 $\mu(U_1 \times U_2) = \mu(U_1)\mu(U_2)$. 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 $\langle \cdot, \cdot \rangle_{H_i}$. Note, that for discrete $\mathcal{X}$ or $\mathcal{Z}$ the respective integrals can be replaced by sums (integral is evaluated by a measure of Dirac delta functions at the discrete points).

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

$$f(x) = (T_k \alpha)(x) = \int_{\mathcal{Z}} k(x, z) \; \alpha(z) \; d\mu(z) , \tag{44}$$

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. (31), with $\alpha_j \;=\; \alpha(z^j)$

$$f(u) \;=\; \sum_{j=1}^{P} \alpha_j \; k\left(u, z^j\right) \;=\; \sum_{j=1}^{P} \alpha_j \; K_{(u)j} \; . \tag{45}$$

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. ℓ^2 denotes the Hilbert space of the set of infinite vectors $\boldsymbol{a} \;=\; (a_1, a_2, \ldots)$, where $\sum_i a_i^2$ converges, with dot product $\langle \boldsymbol{a}, \boldsymbol{b} \rangle_{\ell^2} \;=\; \sum_i a_i b_i$ and the norm $\|\boldsymbol{a}\|_{\ell^2} \;=\; \left(\sum_i a_i^2\right)^{\frac{1}{2}}$. 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, in which the "column" objects are used to describe 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 $T_k : H_1 \rightarrow H_2$

$$\left(T_k \alpha\right)(x) \;=\; f(x) \;=\; \int_{\mathcal{Z}} k(x,z) \; \alpha(z) \; dz \; . \tag{46}$$

Then

$$\|f\|_{H_2}^2 \;\;=\;\; \langle T_k^* T_k \alpha, \alpha \rangle_{H_1} \tag{47}$$

where T_k^* is the adjoint operator of T_k and there exists an expansion

$$k(x,z) \;=\; \sum_n s_n \; e_n(z) \; g_n(x) \tag{48}$$

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.

For $\mathcal{X} \;=\; \mathcal{Z}$ and symmetric, positive definite kernel k, we obtain the eigenfunctions $e_n \;=\; g_n$ of T_k with corresponding eigenvalues s_n.

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

$$\|f\|_{H_2}^2 = \langle T_k\alpha, T_k\alpha\rangle_{H_2} = \langle T_k^* T_k\alpha, \alpha\rangle_{H_1} . \tag{49}$$

The singular value expansion of T_k is

$$T_k\alpha = \sum_n s_n \langle \alpha, e_n\rangle_{H_1} \; g_n \tag{50}$$

(see Werner, 2000, Theorem VI.3.6). The values s_n are the singular values of T_k for the orthonormal systems $\{e_n\}$ on H_1 and $\{g_n\}$ on H_2. We define

$$r_{nm} := \langle T_k e_n, g_m\rangle_{H_2} = \delta_{nm} \; s_m , \tag{51}$$

where the last "=" results from eq. (50) for $\alpha = e_n$. The sum

$$\sum_m r_{nm}^2 = \sum_m \left(\langle T_k e_n, g_m\rangle_{H_2}\right)^2 \leq \|T_k e_n\|_{H_2}^2 < \infty \tag{52}$$

converges because of Bessel's inequality (the $\leq$-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

$$T_k\alpha = \sum_l \langle \alpha, \tilde{e}_l\rangle_{H_1} \; T_k\tilde{e}_l , \tag{53}$$

where we used that T_k is continuous. 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 $\{e_n\}$:

$$\begin{aligned} T_k\alpha &= \sum_n \langle \alpha, e_n\rangle_{H_1} \; T_k e_n \qquad (54)\\ &= \sum_n \langle \alpha, e_n\rangle_{H_1} \left(\sum_m \langle T_k e_n, g_m\rangle_{H_2} \; g_m\right) = \sum_{n,m} r_{nm} \langle \alpha, e_n\rangle_{H_1} \; g_m . \end{aligned}$$

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 $\{e_n \; g_m\}$ are an ONS in H_3 (which can be completed to an ONB) and $\sum_{n,m} r_{nm}^2 < \infty$ (cf. eq. (52)), the kernel

$$\tilde{k}(z,x) := \sum_{n,m} r_{nm} \; e_n(z) \; g_m(x) \tag{55}$$

is from H_3. We observe that the induced Hilbert-Schmidt operator $T_{\tilde{k}}$ is equal to T_k:

$$\left(T_{\tilde{k}}\alpha\right)(x) = \sum_{n,m} r_{nm} \langle \alpha, e_n\rangle_{H_1} \; g_m(x) = (T_k\alpha)(x) , \tag{56}$$

where the first "="-sign follows from eq. (55) and the second "="-sign from eq. (54).

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 from eq. (51) $\langle T_k e_l, g_t \rangle_{H_1} = \delta_{lt}\ s_l$ and $\langle T_k e_l, g_t \rangle_{H_1} = r_{lt}$ from eq. (56), and, therefore, $r_{lt} = \delta_{lt} s_l$. Inserting $r_{nm} = \delta_{nm} s_n$ into eq. (55) proves the eq. (48) in the theorem.

The last statement of the theorem follows from the fact that $|T_k| = (T_k^* T_k)^{1/2} = T_k$ (T_k is positive and selfadjoint) and, therefore, $e_n = g_n$ (Werner, 2000, proof of Theorem VI.3.6, page 246 top).

■

As a consequence of this theorem, for finite $\mathcal{Z}$ 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.

$$\begin{aligned} \boldsymbol{\phi}(x) &:= (s_1 g_1(x), s_2 g_2(x), \ldots)\ , \\ \boldsymbol{\omega}(z) &:= (e_1(z), e_2(z), \ldots)\ , \\ \langle \boldsymbol{\phi}(x), \boldsymbol{\omega}(z) \rangle &= \sum_n s_n\ e_n(z)\ g_n(x) \ = \ k(z,x)\ . \end{aligned} \tag{57}$$

Note, that finite $\mathcal{Z}$ ensures that $\langle \boldsymbol{\omega}(z), \boldsymbol{\omega}(z) \rangle$ converges. 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. (54) we obtain for the classification or regression function

$$f(x) = \sum_n s_n\ \langle \alpha, e_n \rangle_{H_1}\ g_n(x)\ . \tag{58}$$

The classification or regression function is well defined because sets of zero measure vanish through integration in eq. (44), which is confirmed through expansion eq. (58), where the zero measure is "absorbed" in the terms $\langle \alpha, e_n \rangle_{H_1}$.

The expansion of the classification or regression function $f(x)$ into the ONS g_m (cf. eq. (58)) should be ensured to converge absolutely and uniformly in x to justify the analysis in eq. (32), to allow derivatives of g_m with respect to x, and to ensure that $f(x)$ is continuous as a function of x. The latter 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 $\boldsymbol{w} := (\langle \alpha, e_1 \rangle_{H_1}, \langle \alpha, e_2 \rangle_{H_1}, \ldots)$, and $\boldsymbol{\phi}(x) := (s_1 g_1(x), s_2 g_2(x), \ldots)$. 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 fol-

lowing sum convergences absolutely and uniformly:

$$f(x) \;=\; \langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle_{\ell^2} \;=\; \sum_n s_n \; \langle \alpha, e_n \rangle_{H_1} \; g_n(x) \; . \tag{59}$$

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

$$\|\boldsymbol{\phi}(x)\|^2_{\ell^2} \;=\; \sum_n \left(s_n \; g_n(x)\right)^2 \;=\; \sum_n \left((T_k e_n)(x)\right)^2 \tag{60}$$

$$= \; \sum_n \left(\langle k(x,.), e_n \rangle_{H_1}\right)^2 \;\leq\; \|k(x,.)\|^2_{H_1} \;\leq\; \sup_{x \,\in\, \mathcal{X}} \{ \int_{\mathcal{Z}} (k(x,z))^2 \;\; dz \} \;\leq\; K^2 \; ,$$

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 $\|w\|^2_{\ell^2} \;\leq\; \|\alpha\|^2_{H_1}$ we use again Bessel's inequality:

$$\|w\|^2_{\ell^2} \;=\; \sum_n \left(\langle \alpha, e_n \rangle_{H_1}\right)^2 \;\leq\; \|\alpha\|^2_{H_1} \; . \tag{61}$$

Finally, we prove that the sum

$$f(x) \quad = \quad \langle \boldsymbol{w}, \boldsymbol{\phi}(x) \rangle_{\ell^2} \;=\; \sum_n s_n \; \langle \alpha, e_n \rangle_{H_1} \; g_n(x) \tag{62}$$

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

$$\sum_{n=m}^{\infty} \left(\langle \alpha, e_n \rangle_{H_1}\right)^2 \;\leq\; \left(\frac{\epsilon}{K}\right)^2 \tag{63}$$

for $\epsilon > 0$ (because of eq. (61) 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. (60) and (63). Because m is independent of x, the convergence is absolutely and uniformly, too.
■

Eq. (44) or, equivalently, (59) 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 $\mu(x) = \sum_{i=1}^{L} \delta\left(x^i\right)$, $\mu(z) = \sum_{j=1}^{P} \delta\left(z^j\right)$, and $\alpha_j := \alpha\left(z^j\right)$. We obtain

$$\begin{aligned}
f(x) &= \sum_{j=1}^{P} \alpha_j \; k\left(x, z^j\right) \; = \; \left\langle \boldsymbol{\phi}\left(x\right), \sum_{j=1}^{P} \alpha_j \; \boldsymbol{\omega}\left(z^j\right) \right\rangle \; , \\
\boldsymbol{X}_\phi &= \left(\boldsymbol{\phi}\left(x^1\right), \boldsymbol{\phi}\left(x^2\right), \ldots, \boldsymbol{\phi}\left(x^L\right)\right) \; , \\
\boldsymbol{Z}_\omega &= \left(\boldsymbol{\omega}\left(z^1\right), \boldsymbol{\omega}\left(z^2\right), \ldots, \boldsymbol{\omega}\left(z^P\right)\right) \; , \\
\boldsymbol{w} &= \sum_{j=1}^{P} \alpha_i \; \boldsymbol{\omega}\left(z^j\right) \text{ (expansion into support vectors)}, \\
K_{ij} &= \left\langle \boldsymbol{\phi}\left(x^i\right), \boldsymbol{\omega}\left(z^j\right) \right\rangle \; = \; \sum_n s_n \; e_n\left(z^j\right) \; g_n\left(x^i\right) \; = \; k\left(x^i, z^j\right), \\
\boldsymbol{K} &= \boldsymbol{X}_\phi^\top \; \boldsymbol{Z}_\omega \; , \text{ and} \\
\|f\|_{H_2}^2 &= \boldsymbol{\alpha}^\top \boldsymbol{K}^\top \boldsymbol{K} \; \boldsymbol{\alpha} \; = \; \|\boldsymbol{X}_\phi^\top \; \boldsymbol{w}\|_2^2 \text{ (the objective function).} \qquad (64)
\end{aligned}$$

Note, that $\boldsymbol{w}$ is not unique with respect to the subspace which is mapped to zero by the matrix $\boldsymbol{X}_\phi$. 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 also find that the primal problem of the P-SVM (e.g. eq. (25)) corresponds to the formulation in H_2, while the dual (e.g. eq. (30)) corresponds to the formulation in H_1. Primal and dual P-SVM formulations can be transferred into each other via the property $\langle T_k\alpha, T_k\alpha\rangle_{H_2} = \langle T_k^* T_k\alpha, \alpha\rangle_{H_1}$.

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, 2003.

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, 1990.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. B*, 57(1):289–300, 1995.

Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29(4):1165–1188, 2001.

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.

T. G. Dietterich. Approximate statistical test for comparing supervised classification learning algorithms. *Neural Computation*, 10:1895–1923, 1998.

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.

R. Herbrich, T. Graepel, P. Bollmann-Sdorra, and K. Obermayer. Learning a preference relation in IR. In *Proceedings Workshop Text Categorization and Machine Learning, International Conference on Machine Learning 1998*, pages 80–84, 1998.

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. Classification, regression, and feature selection on matrix data. Technical Report 2004/2, Technische Universität Berlin, Fakultät für Elektrotechnik und Informatik, 2004a.

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*, pages 319–355. MIT Press, 2004b.

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

S. Hochreiter and K. Obermayer. Nonlinear feature selection with the potential support vec tor machine. In I. Guyon, S. Gunn, M. Nikravesh, and L. Zadeh, editors, *Feature extraction, Foundations and Applications*. Springer, 2005.

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.

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.

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.

S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pages 41–48. IEEE, 1999.

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.

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, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson. Generalization bounds via eigenvalues of the gram matrix. Technical Report NC2-TR-1999-035, NeuroCOLT2, 1999.

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.

H. D. White and K. W. McCain. Bibliometrics. *Annual Review of Information Science and Technology*, 24:119–186, 1989.