Available at
www.ComputerScienceWeb.com
POWERED BY SCIENCE DIRECT®

Neurocomputing 55 (2003) 5–20

NEUROCOMPUTING

www.elsevier.com/locate/neucom

Advanced support vector machines and kernel methods

V. David Sánchez A.*

Advanced Computational Intelligent Systems Corp., Pasadena, CA 91103, USA

Received 30 March 2002; accepted 13 March 2003

Abstract

Kernel methods (KMs) and support vector machines (SVMs) have become very popular as methods for learning from examples. The basic theory is well understood and applications work successfully in practice. Initially illustrated by their use in classification and regression tasks, recent advanced techniques are presented and key applications are described. Issues of numerical optimization, working set selection, improved generalization, model selection, and parameter tuning are addressed. Application research covering the use of SVMs in text categorization, computer vision, and bioinformatics is discussed.

Keywords: Classification; Kernel method (KM); Neural networks; Pattern recognition; RBF network; Regression; Reproducing kernel Hilbert spaces (RKHS); Support vector machine (SVM); Support vector regression (SVR); Statistical learning theory; Structural risk minimization (SRM)

1. Introduction

Support vector machines (SVMs) and kernel methods (KMs) have become in the last few years one of the most popular approaches to learning from examples with many potential applications in science and engineering. Introductory treatment and some advances of this subject matter have been provided in [7,10,20,67,77,78]. As a learning method, it is often used to train and design radial basis function (RBF) networks. Learning methods for this type of architecture starting with simple training procedures to robust and automatic design methods were presented in [17,41,55,61–64]. Given a

* Corresponding author. Advanced Computational Intelligent Systems Corp., P.O. Box 1424, La Canada, CA 91012, USA.
E-mail address: vdavidsancheza@acis-research-corporation.com (V.D. Sánchez A.).

0925-2312/03/$ - see front matter © 2003 Published by Elsevier B.V.
doi:10.1016/S0925-2312(03)00373-4

set of examples $\{(\vec{x}_i, y_i), \vec{x}_i \in \mathbf{R}^n,\ y_i \in \mathbf{R},\ i = 1,\ldots,N\}$, the SVM learning method in its basic form creates an approximation function $f(\vec{x}) = b + \sum_{j=1}^{m} y_j \cdot \alpha_j \cdot K(\vec{x}_j, \vec{x})$ with $y \approx f(\vec{x})$ for regression and $y \approx \operatorname{sgn} f(\vec{x})$ for dichotomous classification for instance. For that purpose, a subset of support vectors $\{\vec{x}_j, j = 1,\ldots,m\} \subset \{\vec{x}_i,\ i = 1,\ldots,N\}$ is determined, the kernel function K is chosen, and the parameters $b, \alpha_j,\ j = 1,\ldots,m$ are estimated.

KMs are methods that use kernels [2] of the form $K(\vec{x}_1, \vec{x}_2) = \vec{\phi}(\vec{x}_1) \cdot \vec{\phi}(\vec{x}_2), \cdot$ is an inner product and $\vec{\phi}$ is in general a nonlinear mapping from input space X onto feature space Z. KMs are used among others, in SVMs kernel principal component analysis (PCA), kernel Gram–Schmidt, Gaussian processes, and Bayes point machines. The symmetry of the inner product determines the symmetry of the kernel. The necessary and sufficient condition for a symmetric function to be a kernel is to be positive definite [48], thus statistically seen, kernels are covariances. In practice, the kernel function K is directly defined. $\vec{\phi}$ and the feature space Z are implicitly derived from its definition. Kernel substitution of the inner product can be applied for generating SVMs for classification based on margin maximization as we will describe in this paper, to improve generalization [34], or to diminish the number of support vectors in hypothesis construction [76].

Several areas of application research make use of SVM approaches. In the sequel only a sample set is presented to provide the reader with a clear impression of their applicability. For example, handwritten digit recognition using SVMs was proposed in [19,45]. Another example is the use of SVMs for text categorization applications as reported in [23,40]. A SVM-based approach for face detection was introduced in [53]. An emergent field of application for SVMs is in the area of pharmaceutical data analysis and drug design, explicitly for modeling quantitative structure–property relationships and quantitative structure–activity relationships (QSPRs/QSARs), see for instance [6,21]. Modified SVMs and SVM experts have been proposed for time series forecasting [11,75]. SVMs are also used in computational neuroscience, for instance as a computational model for detection of symmetry when eye movement is connected with attention and visual perception as reported in [28].

This paper is subdivided as follows. After introductory remarks in this section, illustrative examples of the use of SVMs are described in Section 2. Section 3 discusses areas of basic research including basic methodology, primal and dual problems, numerical optimization, working set selection, improving generalization, model selection and tuning, and incorporation of a priori knowledge. Section 4 discusses in-depth areas of applied research including text categorization, computer vision, and bioinformatics. Finally, the conclusions are drawn in Section 5.

2. Illustrative examples

Like with other neural network architectures and associated learning methods, historically, the treatment of classification and regression tasks has helped to illustrate their use, but at the same time, to point to key differences. That procedural knowledge is applied in this section to illustrate the use of SVMs and KMs.

2.1. Classification

Given a data set $\{(\vec{x}_i, y_i), \vec{x}_i \in \mathbf{R}^n, y_i \in \{-1, +1\},\ i = 1, \ldots, N\}$. The binary classification problem can be posed as stated in expression (1), see e.g. [78].

$$\begin{aligned} \min_{w,b,\varepsilon} \quad & \mathscr{F} = \frac{1}{2} w^{\mathrm{T}} w + C \sum_{i=1}^{N} \varepsilon_i \\ \text{s.t.} \quad & y_i[w^{\mathrm{T}} \phi(x_i) + b] \geqslant 1 - \varepsilon_i \quad i = 1, \ldots, N, \\ & \varepsilon_i \geqslant 0 \qquad\qquad\qquad\quad i = 1, \ldots, N, \end{aligned} \tag{1}$$

where $y_i[w^{\mathrm{T}} \phi(x_i) + b] \geqslant 1$ comprises first the given constraints $w^{\mathrm{T}} \phi(x_i) + b \geqslant +1$ if $y_i = +1$, $w^{\mathrm{T}} \phi(x_i) + b \leqslant -1$ if $y_i = -1$, and ε_i are the slack variables that allow misclassifications in the set of inequalities. A comparison of approaches to handle the multiclass case as opposed to the binary case is presented in [35].

2.2. Regression

Recently, KMs and SVMs have become very popular for classification and regression, in particular when using RBF networks. Details on support vector algorithms and learning can be found in [20,67,77]. We present an RBF network solution with an associated SVM learning method. Additional, more specific regression-related material can be found e.g. in [22,66,79]. Theoretical foundations were reported in [70]. A mean field approach leads to an efficient iterative learning method for SVM regression as reported in [31]. For regression, the decision function is given in (2).

$$f(\vec{x}) = b + \sum_{i=1}^{m} y_i \cdot (\alpha_i - \beta_i) \cdot K(\vec{x}_i, \vec{x}). \tag{2}$$

For an ε-insensitive loss function:

$$L(x) = \begin{cases} 0, & |x| < \varepsilon, \\ |x|, & \varepsilon \leqslant |x|, \end{cases} \tag{3}$$

a quadratic optimization problem needs to be solved: the dual objective function to be minimized is given in (4) subject to the conditions in (5) $\forall i = 1, \ldots, m$.

$$\begin{aligned} W(\alpha, \beta) = & \sum_{i=1}^{m} y_i(\alpha_i - \beta_i) - \varepsilon \sum_{i=1}^{m} (\alpha_i + \beta_i) \\ & - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} (\alpha_i - \beta_i)(\alpha_j - \beta_j) K(\vec{x}_i, \vec{x}_j), \\ & 0 \leqslant \alpha_i \leqslant C, \ \ 0 \leqslant \beta_i \leqslant C, \end{aligned} \tag{4}$$

$$\sum_{i=1}^{m} \alpha_i = \sum_{i=1}^{m} \beta_i. \tag{5}$$

The bias b is typically determined by averaging individual values which are gained from the Karush–Kuhn–Tucker conditions leading to $b = y_i - \vec{w}.\ \vec{x}_i \pm \varepsilon$, see e.g. [9], with:

$$\vec{w} = \sum_{i=1}^{m} y_i(\alpha_i^* - \beta_i^*)\vec{x}_i, \tag{6}$$

where α_i^* and β_i^* are the optimal values previously determined. Similar quadratic optimization problems are generated when instead of the ε-insensitive loss function quadratic or robust loss functions [33,36] are utilized. A robust learning method for RBF regression networks was introduced in [63]. When the same statistical learning framework is used, the solution provided with the RBF regression network to the nonlinear regression problem shows a close connection to the solution to the linear regression task which follows. The linear regressor's decision function is given in (7). The optimization problem consists in the maximization of the functional given in (8) subject to the constraints in (5) as before, and the solution $\vec{w}$ is given in (9), whereas b is determined as in the case of the RBF network solution.

$$f(\vec{x}) = b + \vec{w} \cdot \vec{x}, \tag{7}$$

$$W(\alpha, \beta) = \sum_{i=1}^{m} y_i(\alpha_i - \beta_i) - \varepsilon \sum_{i=1}^{m} (\alpha_i + \beta_i) \cdots$$

$$- \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} (\alpha_i - \beta_i)(\alpha_j - \beta_j)\vec{x}_i \cdot \vec{x}_j, \tag{8}$$

$$\vec{w} = \sum_{i=1}^{m} (\alpha_i^* - \beta_i^*)\vec{x}_i. \tag{9}$$

3. Areas of basic research

Due to the number and the variety of approaches, we will focus our attention and discussion on a few key ideas to first build a common basis and later generate advancement, leaving other methods as a reference.

3.1. Basic methodology

In the case of SVMs used for dichotomous classification (two-class problems), we use the decision function $\mathrm{sgn}(f)$, see (10). The following parameters need to be determined: the bias b, the number of support vectors m, the support vectors $\vec{x}_i$ and associated y_i values, $i = 1, \ldots, m$, as well as the Lagrangian multipliers α_i. For regression, the decision function we use is given in (11) and the parameters β_i need to be determined in addition. The solution to the above-referenced parameter determination problem utilizes the structural risk minimization (SRM) principle:

$$f(\vec{x}) = b + \sum_{i=1}^{m} y_i \cdot \alpha_i \cdot K(\vec{x}_i, \vec{x}), \tag{10}$$

$$f(\vec{x}) = b + \sum_{i=1}^{m} y_i \cdot (\alpha_i - \beta_i) \cdot K(\vec{x}_i, \vec{x}). \tag{11}$$

3.2. Primal and dual problems

In order to determine model parameters for SVMs, either the original primal problem or its dual counterpart are taken as starting point. Primal and dual problems are defined in the sequel. Applying the SRM principle, e.g. to the dichotomous classification case, the so-called primal problem in (12) subject to the constraints given in (13) and (14) needs to be solved in praxis. The first term accounts for the VC dimension of the learning machine. The second term is an upper bound on the number of misclassifications making use of the slack variables, and $\vec{\xi} = (\xi_1, \xi_2, \ldots, \xi_N)^{\mathrm{T}}$.

$$\min \Phi(\vec{w}, \vec{\xi}) = \frac{1}{2}\, \vec{w}^{\mathrm{T}} \cdot \vec{w} + C \cdot \sum_{i=1}^{N} \xi_i, \tag{12}$$

$$y_i(\vec{w}^{\mathrm{T}} \cdot \vec{\phi}(\vec{x}_i) + b) \geqslant 1 - \xi_i, \quad i = 1, \ldots, N, \tag{13}$$

$$\xi_i \geqslant 0, \quad i = 1, \ldots, N \tag{14}$$

Alternatively, the equivalent optimization problem, the so-called dual problem, needs to be solved, whose solution in the dichotomous classification case—we show the C-SVM form—is the solution of (15) subject to (16) and (17) and is used in (10) to evaluate the decision function. This form evolves among others, by replacing $\vec{w} = \sum_{i=1}^{m} \alpha_i y_i \vec{x}_i$, $\alpha_i \neq 0$ only for the support vectors $\vec{x}_i$, $i = 1, \ldots, m$.

$$\min W(\vec{\alpha}) = -\sum_{i=1}^{m} \alpha_i + \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \alpha_i \alpha_j y_i y_j K(\vec{x}_i, \vec{x}_j), \tag{15}$$

$$\sum_{i=1}^{m} \alpha_i y_i = 0, \tag{16}$$

$$0 \leqslant \alpha_i \leqslant C \quad \forall_i = 1, \ldots, m. \tag{17}$$

The dual problem can also be expressed in vectorial form as

$$\min W(\vec{\alpha}) = -\vec{\alpha}^{\mathrm{T}} \cdot \vec{1} + \tfrac{1}{2}\, \vec{\alpha}^{\mathrm{T}} \cdot \mathbf{Q} \cdot \vec{\alpha}, \tag{18}$$

$$\vec{\alpha}^{\mathrm{T}} \cdot \vec{y} = 0, \tag{19}$$

$$0 \leqslant \vec{\alpha} \leqslant \vec{C} \tag{20}$$

with $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)^{\mathrm{T}}$, $\vec{y} = (y_1, y_2, \ldots, y_m)^{\mathrm{T}}$, $\vec{C} = (C, C, \ldots, C)^{\mathrm{T}}$, $C \in \mathbf{R}$, $Q = (Q_{ij})_{i,j=1,\ldots,m}$, $Q_{ij} = y_i y_j K(\vec{x}_i \vec{x}_j)$. Examples of RKHS kernel functions $K(\vec{x}, \vec{y})$ (vector arguments) or $K(x, y)$ (scalar arguments) are summarized in Table 1, cf. [80].

Table 1
Examples of kernel functions

Kernel function	Used in
$\tanh(\vec{x}\cdot\vec{y}-\theta)$	Multilayer perceptron (MLP)
$\exp(-\|\vec{x}-\vec{y}\|^2)$	Gaussian RBF Network
$(\|\vec{x}-\vec{y}\|^2 \pm c^2)^{-1/2}$	Direct/inverse multiquadric
$(1+\vec{x}\cdot\vec{y})^d$	Polynomial of degree d
$\frac{\sin(d+1/2)(x-y)}{\sin(x-y)/2}$	Trigonometric polynomial of degree d
$\|\vec{x}-\vec{y}\|^{2n+1}$,	Thin plate
$\|\vec{x}-\vec{y}\|^{2n}\ln(\|\vec{x}-\vec{y}\|)$	Splines
$B_{2n+1}(x-y)$	B-Splines
$\frac{\sigma}{\pi}\operatorname{sinc}\left[\frac{\sigma}{\pi}(x-y)\right]$	Band-limited Paley Wiener space

Table 2
Examples of loss functions

Loss function	Used in
$(x-y)^2$	Regularization
$\|x-y\|_\varepsilon$	Regression
$\|1-xy\|_+$	Classification
$\theta(1-xy)$	Classification: hard margin
$\theta(-xy)$	Classification: misclassification

Examples of loss functions are summarized in Table 2. Further elaboration is needed to derive the solution to related, but more specific problems. For instance, the primal and corresponding dual problem for C-SVMs [78] and v-SVMs [69] are summarized in Table 3, their relationship is investigated in [13].

3.3. Numerical optimization

Methods to solve the corresponding optimization problem to SVM learning include sequential minimal optimization (SMO) and derivatives [30,56] decomposition methods [44,72], and methods to solve the least-squares SVM formulations [12,42,74]. Software packages that can be used, have been made available to the research community, see e.g. svmlight [39], mysvm [60], libsvm [14], and svmtorch [18].

In the case of regression, the function to be minimized has the form given in (21), for conditions see previous subsection on regression.

$$W(\alpha,\beta)=\sum_{i=1}^{m} y_i(\alpha_i-\beta_i)-\varepsilon\sum_{i=1}^{m}(\alpha_i+\beta_i)$$

Table 3
Primal and dual problems

SVM	Primal problem	Dual problem
C-SVM	$\min \frac{1}{2}\vec{w}^{\mathrm{T}}\vec{w} + C\sum_{i=1}^{N}\xi_i$ $y_i(\vec{w}^{\mathrm{T}}\phi(\vec{x}_i)+b) \geqslant 1-\xi_i$ $\xi_i \geqslant 0,\ i=1,\ldots,N$	$\min \frac{1}{2}\alpha^{\mathrm{T}}\mathbf{Q}\alpha - \vec{1}^{\mathrm{T}}\vec{\alpha}$ $\vec{y}^{\mathrm{T}}\vec{\alpha}=0,\ 0\leqslant\alpha_i\leqslant C,\ i=1,\ldots,m$
v-SVM	$\min \frac{1}{2}\vec{w}^{\mathrm{T}}\vec{w} - v\rho + \frac{1}{N}\sum_{i=1}^{N}\xi_i$ $y_i(\vec{w}^{\mathrm{T}}\phi(\vec{x}_i)+b) \geqslant \rho-\xi_i$ $\xi_i \geqslant 0,\ i=1,\ldots,N,\rho\geqslant 0$	$\min \frac{1}{2}\alpha^{\mathrm{T}}\mathbf{Q}\alpha$ $\vec{y}^{\mathrm{T}}\vec{\alpha}=0,\ \vec{1}^{\mathrm{T}}\vec{\alpha}\geqslant v$ $0\leqslant\alpha_i\leqslant 1/m,\ i=1,\ldots,m$

$$-\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}(\alpha_i-\beta_i)(\alpha_j-\beta_j)K(\vec{x}_i,\vec{x}_j). \tag{21}$$

In general, numerical decomposition methods and the method of nearest points, for solving the respective dual problem are reported in [39,43,52,56,65]. As an example of a more specific method, a decomposition method for training v-support vector classifiers is described in [13]. In decomposition methods, in each iteration the index set of variables is separated into two subsets B and N, B is the index for the working set. Variables corresponding to the indices in N are fixed, while the subproblem on the variables corresponding to the indices in B is suboptimized, thus the last two terms of the objective function in (23) are fixed and can be eliminated from the minimization.

$$\vec{\alpha}=(\vec{\alpha}_B^{\mathrm{T}},\vec{\alpha}_N^{\mathrm{T}})^{\mathrm{T}},\quad \vec{y}=(\vec{y}_B^{\mathrm{T}},\vec{y}_N^{\mathrm{T}})^{\mathrm{T}},\quad \mathbf{Q}=\begin{pmatrix}\mathbf{Q}_{BB} & \mathbf{Q}_{BN}\\ \mathbf{Q}_{NB} & \mathbf{Q}_{NN}\end{pmatrix}, \tag{22}$$

$$\min W(\alpha)=\tfrac{1}{2}\vec{\alpha}_B^{\mathrm{T}}\mathbf{Q}_{BB}\vec{\alpha}_B-\vec{\alpha}_B(\vec{1}-\mathbf{Q}_{BN}\vec{\alpha}_N)+\tfrac{1}{2}\vec{\alpha}_N^{\mathrm{T}}\mathbf{Q}_{NN}\vec{\alpha}_N-\vec{\alpha}_N\vec{1}, \tag{23}$$

$$\vec{\alpha}_B^{\mathrm{T}}\vec{y}_B+\vec{\alpha}_N^{\mathrm{T}}\vec{y}_N=0, \tag{24}$$

$$0\leqslant\vec{\alpha}\leqslant\vec{C}. \tag{25}$$

Pair approaches expand the base set so that all elements are pairs. A reason for it is to keep $\alpha_i\alpha_i^*=0,\ i=1,\ldots,N,\ \alpha_i$ is the Lagrange multiplier associated to $\vec{x}_i$ in a given data set $\{(\vec{x}_i,y_i),\ i=1,\ldots,N\}$, and to keep the number of nonzero variables small. Recent analysis has shown [46] that pair approaches for regression offer almost no or only marginal advantage regarding convergence, i.e., number of iterations, with respect to base approaches. Base approaches use the base set as working set.

3.4. Working set selection

One approach of selecting the working set is by identifying elements that violate the KKT condition, e.g. [52,65]. A systematic approach for working set selection [39,82], here applied to v-support vector classifiers, is based on the solution of the following problem:

$$\min(\vec{\nabla} W(\vec{\alpha}^{\mathrm{T}}))^{\mathrm{T}}\vec{d}, \tag{26}$$

$$\vec{y}^{\,\mathrm{T}}\vec{d} = 0, \tag{27}$$

$$-\vec{1} \leqslant \vec{d} \leqslant \vec{1}, \tag{28}$$

$$d_i \geqslant 0, \quad \forall i : \alpha_i = 0, \tag{29}$$

$$d_i \leqslant 0, \quad \forall i : \alpha_i = C, \tag{30}$$

$$|\{d_i : d_i \neq 0\}| \leqslant q. \tag{31}$$

Up to q variables are involved in the direction $\vec{d}$ of descent, $(\vec{\nabla} W(\vec{\alpha}^{\mathrm{T}}))^{\mathrm{T}}$ is the vector of partial derivatives. The rest of the conditions translate in essence directly from condition (19) of the dual optimization problem.

3.5. Improving generalization

One idea to improve the generalization capabilities of SVM classifiers is to enhance the spatial resolution on the boundary surface. The idea is based on introducing a conformal mapping into the Riemannian geometry induced by the classifier kernel function [1], for similar differential geometrical enhancements see [8,51]. Following the definition of an example-based learning classifier and its SVM solution: given the example set $\{(\vec{x}_i, y_i),\ i=1,\ldots,N\}$, the non-linear mapping $\vec{\phi}$ of the input space $X \subset \mathbf{R}^n$ into feature space $Z \subset \mathbf{R}^m$ by the SVM is carried out according to (32), classification is done based on the sign of the linear discriminant function g according to (33) in feature space with a nonlinear boundary $f(\vec{x}) = 0$ in input space, and finally the SVM solution is computed by applying structure risk minimization, i.e., minimizing an upper bound of the generalization error, leading to the maximization of the margin between the data and the separating hyperplane defined in (34).

$$\vec{z} = \vec{\phi}(\vec{x}), \quad \vec{x} \in X,\ \vec{z} \in Z, \tag{32}$$

$$f(\vec{x}) = g(\vec{z}) = \vec{w} \cdot \vec{z} + b, \quad b \in \mathbf{R}, \tag{33}$$

$$\max \frac{2}{\|\vec{w}\|}. \tag{34}$$

The transformation $\vec{\phi}$ induces a Riemannian metric under certain conditions, e.g., Z is a Hilbert space. For a reproducing kernel function $K(\vec{x}_1, \vec{x}_2) = \vec{\phi}(\vec{x}_1) \cdot \vec{\phi}(\vec{x}_2), \cdot$ being

the inner product, the magnification factor $\sqrt{h}$ is of interest for this analysis, where h is a function that helps define the elements of the Riemannian metric tensor H in (35) and the corresponding metric defined according to (36). The magnification factor directly affects how areas of input space X are magnified in the feature space Z under the mapping ϕ and can be determined from the kernel function K. As an example, the metric derived for the popular Gaussian radial kernel—typically used in RBF networks [64]—is shown in (37).

$$H = (h_{ij}(\vec{x}))_{i,j=1,\ldots,n}, \tag{35}$$

$$h_{ij}(\vec{x}) = \left(\frac{\partial}{\partial x_i}\vec{\phi}(\vec{x})\right) \cdot \left(\frac{\partial}{\partial x_j}\vec{\phi}(\vec{x})\right), \tag{36}$$

$$h_{ij}(\vec{x}) = \frac{1}{\sigma^2}\delta_{ij}, \quad K(\vec{x}_1,\vec{x}_2) = f\left(\tfrac{1}{2}\|\vec{x}_1 - \vec{x}_2\|^2\right). \tag{37}$$

Training using the improved SVM classifier transforms into a two-step approach. The first step is training using the original kernel K. The second step is training using the modification of K to K' using a conformal transformation of the kernel by a factor $c(\vec{x})$, the positive scalar function in (38), according to (39). The improvement in class separability is achieved by increasing the margin in (34) leading to a modification of the metric $h_{ij}(\vec{x})$ around the boundary defined by $f(\vec{x}) = 0$ according to (40) in the case of the Gaussian RBF kernel.

$$c(\vec{x}) = \sum_{i=1}^{p} w_i \mathrm{e}^{-\frac{\|\vec{x}-\vec{x}_i\|^2}{2\tau^2}}, \tag{38}$$

$$K'(\vec{x}_1,\vec{x}_2) = c(\vec{x}_1)c(\vec{x}_2)K(\vec{x}_1,\vec{x}_2), \tag{39}$$

$$h'_{ij}(\vec{x}) = \frac{\partial c(\vec{x})}{\partial x_i}\frac{\partial c(\vec{x})}{\partial x_j} + c^2(\vec{x})h_{ij}(\vec{x}). \tag{40}$$

Some preliminary experimental results have strengthened the validity of these information-geometrical considerations as an approach to optimal, data-dependent SVM kernel choice and generalization improvement. Another example of an approach for improving generalization when SVM learning is applied to RBF networks is described in [81]. The theoretical foundations applied to determine the optimal spread parameter for a Gaussian kernel in classification and regression problems are based on Fisher discrimination and scale space theory respectively.

3.6. Model selection and tuning

The relevance of tuning model parameters, e.g., of a SVM solving a classification problem, can be made plausible directly, because they affect the classifier's performance, i.e., they help minimize its generalization error. The subject of model selection and parameter tuning can be better understood when conceived as being more closely

related to the learning problem goal itself, i.e., optimizing the generalization capability or minimizing the generalization error. Note that in contrast to it, training optimizes the values of certain model parameters keeping the values of other key parameters, sometimes called hyperparameters, fixed.

To be more specific, let us briefly review the due optimization problem to be solved after posing the SVM-based solution to binary classification using RBF kernels with $K(\vec{x}_i,\vec{x}_j)=\mathrm{e}^{-\gamma\|\vec{x}_i-\vec{x}_j\|^2}$ as stated in (41).

$$\begin{aligned} &\max_{\alpha} \quad \left\{\sum_{i=1}^{n_p}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{n_p}\alpha_i\alpha_j y_i y_j K(\vec{x}_i,\vec{x}_j)\right\},\\ &\text{s.t.} \quad 0\leqslant\alpha_i\leqslant c,\forall i,\\ &\qquad \sum_{i=1}^{n_p} y_i\alpha_i=0. \end{aligned} \tag{41}$$

In this case, training delivers optimal values for the Lagrange multipliers α_i while model selection and hyperparameter tuning delivers optimal values for the error-weighting regularization factor c and the RBF kernel parameter γ. The regularizing character of c can be directly seen in the objective function of the primal optimization problem. c and γ decisively affect the classification performance. The procedure according to the penalization by maximal discrepancy approach to hyperparameter tuning and effective VC dimension determination in α-parameter space is given in (42) and (43). Bounds for a specific performance measure, the leave-one-out error, were given in [15,16]. Performance measures used for model selection and SVM hyperparameter tuning include k-fold cross-validation, leave-one-out (LOO), Xi-alpha bound, generalized approximate cross-validation (GACV), approximate span bound, Vapnik Chervonenkis (VC) bound, and radius-margin bound. These measures are compared in [24]. Other relevant issues of model selection and error estimation are discussed in [3].

$$\max_{\alpha}(v_2-v_1), \quad c,\gamma \text{ fixed}, \tag{42}$$

$$v_i=\frac{1}{n_p/2}\sum_i l(f(\vec{x}_i),y_i), \tag{43}$$

$$l(f(\vec{x}_i),y_i)=\begin{cases}1, & f(\vec{x}_i)\cdot y_i\leqslant 0,\\ 0, & \text{otherwise.}\end{cases}$$

3.7. Incorporation of a priori knowledge

One straightforward way of incorporating a priori knowledge into the SVM learning framework is by replacing the bias term b in the objective function for SVMs by a parametric model consisting of a linear combination of functions according to (44) for regression, leading to semi-parametric models [71]. A priori knowledge can therefore be

incorporated into a semiparametric formulation of SVMs, e.g., by means of a low-order polynomial. Other frameworks which allow the inclusion of a priori knowledge are presented in [68,80].

$$f(\vec{x}) = \sum_{i=1}^{m} b_i f^i(\vec{x}) + \sum_{i=1}^{m} \alpha_i \cdot K(\vec{x}_i, \vec{x}). \tag{44}$$

4. Areas of applied research

Techniques based on SVMs and KMs are used to solve problems in different areas of applied research including problems of text categorization, computer vision, bioinformatics, analysis of financial and biological data, time series prediction, and high energy physics. The three first areas of application are covered in more depth in the sequel.

4.1. Text categorization

The task of natural language text categorization consists in classifying documents into a number of predefined categories based on their content. The documents can be in none, in one, or in multiple categories. Attribute value representation of text is used to adequately represent the document text, i.e., character strings, for learning and classification. Each distinct word in a document represents a feature, the number of appearances corresponds to its value. This representation scheme leads to a space of very high dimension space of 10 k dimensions and more. Because of this space dimensionality, the dense concept vector, and the sparse instance vectors, SVMs are expected to perform well for this application. As experiments show, SVMs outperform other methods for this task [40] including a naive Bayes classifier [38], the Rocchio algorithm [58], a distance weighted k-nearest neighbor [49], and the C4.5 decision tree/rule learner [57]. In addition, no model parameter tuning is necessary. Performance measures in text categorization include recall, precision, error rate, false alarm, and miss rate. Email spam categorization using Ripper, Rocchio, boosting decision trees, and SVMs is discussed in [23]. A comparison of five algorithms for text categorization is reported in [26] including find similar, decision trees, naive Bayes, Bayes nets, and SVMs. The linear SVM accuracy for the Reuters-21578 collection is one of the best reported, the model is simple, uses 300 binary features per category and the SMO training algorithm.

4.2. Computer vision

Object detection in computer vision where concepts like faces are involved requires novel approaches. The main reason is that these concepts cannot be expressed in terms of a small and meaningful set of features. Pattern variations in face detection, difficult to parameterize analytically, include facial appearance, expression, light and shadow distribution. The only feasible solution approach to these problems demands methods

that can deal with large data sets $O(10^6)$, high dimensions $O(10^3)$, and essentially learn the solution from a set of examples. A SVM based approach to face detection was introduced in [53]. Given a digitized video image or scanned photograph, the task consists of determining whether there are human faces on it and if yes, their location encoding. The SVM based face detection system performs comparably to the state of the art algorithms including the ones reported in [59] and [73].

Another computer vision application of SVMs is a pattern classification approach to dynamical object detection in video sequences [54]. The object class is described by an overcomplete set of Haar wavelet features. No feature selection from the dictionary is carried out. The model is learned in the full feature space of $O(10^3)$ dimension. Compared with previous approaches that use Kalman filters or hidden Markov models to explicitly model the object dynamics, see e.g. [4], the SVM based approach does not assume any a priori dynamics model and reduces transient false positives. Several object classes can be used including people for which experimentation was reported. Five consecutive frames, each of 128×64 patterns, leads to a single 6630-dimensional feature. The training set used consisted of 1379 and 3822 positive and negative examples respectively. As opposed to classical classification techniques that would overfit the high-dimensional space, the SVM approach controls simultaneously the training error and the classifier complexity. The system learns the salient physical structure and the people's dynamics present in the video sequences.

4.3. Bioinformatics

Gene expression analysis performed by SVMs is discussed in [5]. Gene expression data from DNA microarray hybridization experiments was used. The gold standard applied is MYGD (MIPS yeast genome database) [50] classification. SVMs with different similarity metrics as well as other machine learning approaches are used in the analysis including Parzen windows, Fisher's linear discriminant, and two decision tree classifiers. SVMs showed superior performance in accurately classifying genes into functional categories with respect to the other methods. Among different types of SVM, the RBF SVMs performed the best for this task. Previous work [29] had used a normalized dot product as similarity metric. Apart from large amounts of mRNA expression data, other sources of information about the genes can be used including the presence of transcription factor binding sites in the promoter region or sequence features of the translated protein. The method shows potential to cope with the more complex problem of reconstructing complete regulatory pathways within the cell.

Detection of remote protein homologies by SVMs is discussed in [37]. The discriminative method is built on top of a generative model, e.g. hidden Markov models (HMMs), built from multiple sequences and which provides appropriate features for the identification of structural relationships. Accordingly, the protein sequences are mapped to points of an Euclidian feature space with fixed dimension. The method provides with a significant improvement in relation to previous approaches to classification of protein domains based on remote protein homologies including approaches based on training of HMM parameters [27,47], based on neural discrimination [25], and family pairweise search homology methods [32].

5. Conclusions

Advanced support vector machines (SVMs) and kernel methods (KMs) were presented including methods that incorporate generalization improvement, model selection, hyperparameter tuning as well as a priori knowledge. The basic methodology was presented first to provide a common basis including issues of primal/dual problems and numerical optimization as well. SVMs and KMs can be in general used for classification, regression, clustering, density estimation, and novelty detection. Areas of application research discussed included text categorization, computer vision, and bioinformatics. The elegance of the formalisms involved and their successful use in diverse science and engineering applications confirm the expectations raised in this appealing learning from examples approach. On the other hand, open issues reaffirm the due commitment to their further development and investigation.

References

[1] S. Amari, S. Wu, Improving support vector machine classifiers by modifying kernel functions, Neural Networks 12 (6) (1999) 783–789.
[2] N. Aronszajn, Theory of reproducing kernels, Trans. Am. Math. Soc. 68 (1950) 337–404.
[3] P. Bartlett, S. Boucherou, G. Lugosi, Model selection and error estimation, Mach. Learn. 48 (1–3) (2002) 85–113.
[4] C. Bregler, Learning and recognizing human dynamics in video sequences, in: R. Nevatia, G. Medioni (Eds.), IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society Press, Silver Spring, MD, 1997, pp. 568–574.
[5] M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C.W. Sugnet, T.S. Furey, M. Ares Jr., D. Haussler, Knowledge-based analysis of microarray gene expression data by using support vector machines, Proc. Natl. Acad. Sci. 97 (2000) 262–267.
[6] R. Burbidge, M. Trotter, B. Buxton, S. Holden, Drug design by machine learning: support vector machines for pharmaceutical data analysis, Comput. Chem. 26 (1) (2001) 5–14.
[7] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Mining Knowledge Discovery 2 (2) (1998) l21–167.
[8] C.J.C. Burges, Geometry and invariance in kernel based methods, in: B. Schoelkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, The MIT Press, Cambridge, MA, 1999, pp. 89–116.
[9] C. Campbell, An introduction to kernel methods, in: R.J. Howlett, L.C. Jain (Eds.), Radial Basis Function Networks: Design and Applications, Physica Verlag, Wurzburg, 2000, pp. 155–192.
[10] C. Campbell, Kernel methods: a survey of current techniques, Neurocomputing 48 (1–4) (2002) 63–84.
[11] L. Cao, Support vector machines experts for time series forecasting, Neurocomputing 51 (2003) 321–339.
[12] G.C. Cawley, N.L.C. Talbot, Improved sparse least-squares support vector machines, Neurocomputing 48 (1–4) (2002) 1025–1031.
[13] C.-C. Chang, C.-J. Lin, Training *v*-support vector classifiers: theory and algorithms, Neural Computation 13 (9) (2001) 2119–2147.
[14] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, http://www.csie.ntu.edu.tw/∼cjlin/libsvm/,2003.
[15] O. Chapelle, V.N. Vapnik, Model selection for support vector machines, in: S. Solla, T.K. Leen, K.-R. Muller (Eds.), Advances in Neural Information Processing Systems, Vol. 12, The MIT Press, Cambridge, MA, 2000, pp. 230–236.

[16] O. Chapelle, V.N. Vapnik, O. Bousquet, S. Mukherjee, Choosing multiple parameters for support vector machines, Mach. Learn. 46 (1–3) (2002) 131–159.
[17] S. Chen, C.F.N. Cowan, P.M. Grant, Orthogonal least squares learning algorithm for radial basis function networks, IEEE Trans. Neural Networks 2 (1991) 302–309.
[18] R. Collobert, S. Bangio, SVMTorch: a support vector machine for large-scale regression and classification problems, J. Mach. Learn. Res. 1 (2001) 143–160.
[19] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (1995) 273–297.
[20] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge Univereity Press, Cambridge, 2000.
[21] R. Czerminski, A. Yasri, D. Hartsough, Use of support vector machines in pattern classification: application to QSAR studies, Quant. Struct.-Act. Relat. 20 (2001) 227–240.
[22] H. Drucker, C.J.C. Burges, L. Kaufman, A. Smola, V. Vapnik. Support vector regression machines, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, Vol. 9, The MIT Press, Cambridge, MA, 1997, pp. 155–161.
[23] H. Drucker, D. Wu, V. Vapnik, Support vector machines for spam categorization, IEEE Trans. Neural Networks 10 (5) (1999) 1048–1054.
[24] K. Duan, S.S. Keerthi, A.N. Poo, Evaluation of simple performance measures for tuning SVM hyperparameters, Neurocomputing 51 (2003) 41–59.
[25] I. Dubchak, I. Muchnik, S. Kim, Protein folding class predictor for SCOP: approach based on global descriptors, in: Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology, Halkidiki, Greece, June 1997, pp. 104–108.
[26] S. Dumais, J. Platt, D. Heckerman, M. Sahami, Inductive learning algorithms and representations for text categorization, in: Seventh International Conference on Information and Knowledge Management, Bethesda, MD, USA, 1998.
[27] S.R. Eddy, G. Mitchison, R. Durbin, Maximum discrimination hidden Markov models of sequence consensus, J. Comput. Biol. 2 (1995) 9–23.
[28] H. Eghbalnia, A. Assadi, An application of support vector machines and symmetry to computational modeling of perception through visual attention, Neurocomputing 38–40 (2001) 1193–1201.
[29] M. Eisen, P. Spellman, P. Brown, D. Botstein, Cluster analysis and display of genome-wide expression patterns, Proc. Natl. Acad. Sci. USA 95 (1998) 14863–14868.
[30] G.W. Flake, S. Lawrence, Efficient SVM regression training with SMO, Mach. Learn. 46 (2002) 271–290.
[31] J.B. Gao, S.R. Gunn, C.J. Harris, Mean field methods for the support vector machine regression, Neurocomputing 50 (1–4) (2003) 391–405.
[32] W.N. Grundy, Family-based homology detection via pairwise sequence comparison, in: International Conference on Computational Molecular Biology (RECOMB-98), ACM Press, New York, 1998.
[33] F.R. Hampel, E.M. Rochetti, P.J. Rousseeuw, W.A. Stahel, Robust Statistics, Wiley, New York, 1986.
[34] R. Herbrich, T. Graeppel, C. Campbell, Bayesian learning in reproducing kernel Hilbert spaces, Technical University of Berlin, Department of Computer Science, Technical Report TR 99-11, July 1999.
[35] C.-W. Hsu, C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Networks 13 (2) (2002) 415–425.
[36] P.J. Huber, Robust Statistics, Wiley, New York, 1981.
[37] T. Jaakkola, M. Diekhans, D. Haussler, A discriminative framework for detecting remote protein homologies, J. Computat. Biol. 7 (1–2) (2000) 95–114.
[38] T. Joachims, A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization, in: International Conference on Machine Learning (ICML), Nashville, TN, USA, 1997.
[39] T. Joachims, Making large-scale SVM learning practical, in: B. Schoelkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, The MIT Press, Cambridge, MA, 1998.
[40] T. Joachims, Text categorization with support vector machines: learning with many relevant features, in: Proceedings of the European Conference on Machine Learning (ECML), Chemnitz, Germany, 1998.
[41] P.A. Jokinen, A nonlinear network model for continuous learning, Neurocomputing 3 (4) (1991) 157–176.

[42] S.S. Keerthi, S.K. Shevade, SMO algorithm for least-squares SVM formulations, Neural Comput. 15 (2) (2003) 487–507.
[43] S. Keerthi, C.B.S.K. Shevade, K.R.K. Murphy, A fast iterative nearest point algorithm for support vector machine classifier design, IEEE Trans. on Neural Networks 11 (1) (2000) 124–136.
[44] P. Laskov, An improved decomposition algorithm for regression support vector machines, Mach. Learn. 46 (2002) 315–350.
[45] Y. Le Cun, L.D. Jackel, L. Bottou, C. Cortes, J. Denker, H. Drucker, I. Guyon, U.A. Muller, E. Sackinger, P. Simard, V. Vapnik, Learning algorithms for classification: a comparison on handwritten digit recognition, in: J.H. Kwon, S. Cho (Eds.), Neural Networks: The Statistical Mechanics Perspective, World Scientific, Singapore, 1995, pp. 261–276.
[46] S.-P. Liao, H.-T. Liu, C.-J. Lin, A note on the decomposition methods for support vector regression, Neural Comput. 14 (2002) 1267–1281.
[47] H. Mamitsuka, A learning method of hidden Markov models for sequence discrimination, J. Comput. Biol. 3 (3) (1996) 361–373.
[48] J. Mercer, Functions of positive and negative type and their connection with the theory of integral equations, Philos. Trans. R. Soc. London A 209 (1909) 415–446.
[49] T. Mitchell, Machine Learning, McGraw-Hill, New York, 1997.
[50] MYGD, Munich information center for protein sequences yeast gene database, http://www.mips.biochem.mpg.de/proj/yeast, 1999.
[51] I. Okamoto, S.-I. Amari, K. Takeuchi, Asymptotic theory of sequential estimation: differential geometrical approach, Ann. Stat. 19 (1991) 961–981.
[52] E. Osuna, R. Freund, F. Girosi, An improved training algorithm for support vector machines, in: Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing, Amelia Island Plantation, FL, USA, September 1997, pp. 276–285.
[53] E. Osuna, R. Freund, F. Girosi, Training support vector machines: An application to face detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR'97, Puerto Rico, June 1997, pp. 130–136.
[54] C. Papageorgiou, T. Poggio, A pattern classification approach to dynamical object detection, in: Proceedings of the International Conference on Computer Vision, ICCV, Corfu, Greece, 1999, pp. 1223–1228.
[55] J. Platt, A resource-allocation network for function interpolation, Neural Computation 3 (1991) 213–225.
[56] J.C. Platt, Fast training of support vector machines using sequential minimal optimization, in: B. Schoelkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, The MIT Press, Cambridge, MA, 1998.
[57] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, Los Altos, CA, 1993.
[58] J. Rocchio, Relevance feedback in information retrieval, in: G. Salton (Ed.), The SMART Retrieval System: Experiments in Automatic Document Processing, Prentice-Hall, Englewood Cliffs, NJ, 1979, pp. 313–323.
[59] H.A. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1) (1998) 23–38.
[60] S. Rueping, mySVM: another one of those support vector machines, http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM,2003.
[61] V.D. Sánchez A., On the design of a class of neural networks, J. Network Comput. Appl. 19 (1996) 111–118.
[62] V.D. Sánchez A., Advances towards the automatic design of RBF networks, Int. J. Knowledge-Based Intell. Eng. Syst. 1(3) (1997) 168–174.
[63] V.D. Sánchez A., New robust learning method, Int. J. Smart Eng. Syst. Design 1 (1998) 223–233.
[64] V.D. Sánchez A., Special issue on RBF networks, Neurocomputing 19/20 (1998) 1–3.
[65] C. Saunders, M.O. Stitson, J. Weston, L. Bottou, B. Schoelkopf, A.J. Smola, Support vector machine manual, Technical Report CSD-TR-98-03, Royal Holloway College, 1998.
[66] B. Schoelkopf, P. Bartlett, A.J. Smola, R. Williamson, Shrinking the tube: A new support vector regression algorithm, in: M.S. Kearns, S.A. Solla, D.A. Cohn (Eds.), Advances in Neural Information Processing Systems, Vol. 11, The MIT Press, Cambridge, MA, 1999, pp. 330–336.

[67] B. Schoelkopf, C.J.C. Burges, A.J. Smola (Eds.), Advances in Kernel Methods—Support Vector Learning, The MIT Press, Cambridge, MA, 1999.
[68] B. Schoelkopf, P.Y. Simard, A.J. Smola, V.N. Vapnik, Prior knowledge in support vector kernels, in: M.I. Jordan, M.J. Kearns, S.A. Solla (Eds.), Advances in Neural Information Processing Systems, Vol. 10, The MIT Press, Cambridge, MA, 1998, pp. 640–646.
[69] B. Schoelkopf, A.J. Smola, R. Williamson, P.L. Bartlett, New support vector algorithms, Neural Computation 12 (2000) 1207–1245.
[70] J. Shawe-Taylor, S. Ben-David, P. Koiran, R. Schapire, Special issue on theoretical analysis of real-valued function classes, Neurocomputing 29 (1999) 1–3.
[71] A. Smola, T. Friess, B. Schoelkopf, Semiparametric support vector and linear programming machines, in: M. Kearns, S. Solla, D. Cohn (Eds.), Advances in Neural Information Processing Systems, Vol. 11, The MIT Press, Cambridge, MA, 1999, pp. 585–591.
[72] A.J. Smola, B. Schoelkopf, A tutorial on support vector regression, NeuroColt Technical Report TR-1998-030, Royal Holloway College, 1998.
[73] K.K. Sung, T. Poggio, Example-based learning for view-based human face detection, IEEE Trans. Pattern Anal. Mach. Intell. 20 (1) (1998) 39–51.
[74] J.A.K. Suykens, J. de Brabanter, L. Likas, J. Vandewalle, Weighted least squares support vector machines: robustness and sparse approximation, Neurocomputing 48 (1–4) (2002) 85–105.
[75] F.E.H. Tay, L.J. Cao, Modified support vector machines in financial time series forecasting, Neurocomputing 48 (1–4) (2002) 847–861.
[76] M. Tipping, The relevance vector machine, in: S. Solla, T.K. Leen, K.-R. Muller (Eds.), Advances in Neural Information Processing Systems, Vol. 12, The MIT Press, Cambridge, MA, 2000.
[77] V. Vapnik, The Nature of Statistical Learning Theory, Springer, Berlin, 1995.
[78] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[79] V. Vapnik, S.E. Golowich, A. Smola, Support vector method for function approximation, regression estimation, and signal processing, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, Vol. 9, The MIT Press, Cambridge, MA, 1997, pp. 281–287.
[80] S. Vijayakumar, H. Ogawa, RKHS-based functional analysis for exact incremental learning, Neurocomputing 29 (1–3) (1999) 85–113.
[81] W. Wang, Z. Xu, W. Lu, X. Zhang, Determination of the spread parameter in the Gaussian kernel for classification and regression, Neurocomputing, in press.
[82] G. Zoutendijk, Methods of Feasible Directions: a study in linear and non-linear programming, Elsevier, Amsterdam, 1970.