Kernel Methods: A Survey of Current Techniques

Colin Campbell

Department of Engineering Mathematics, Bristol University,
Bristol BS8 1TR, United Kingdom

Abstract: Kernel Methods have become an increasingly popular tool for machine learning tasks involving classification, regression or novelty detection. They exhibit good generalisation performance on many real-life datasets and the approach is properly motivated theoretically. There are relatively few free parameters to adjust and the architecture of the learning machine does not need to be found by experimentation. In this tutorial we survey this subject with a principal focus on the most well-known models based on kernel substitution, namely, Support Vector Machines.

1 Introduction.

Support Vector Machines (SVMs) have been successfully applied to a number of applications ranging from particle identification, face identification and text categorisation to engine knock detection, bioinformatics and database marketing [9]. The approach is systematic and properly motivated by statistical learning theory [42]. Training involves optimisation of a convex cost function: there are no false local minima to complicate the learning process. The approach has many other benefits, for example, the model constructed has an explicit dependence on the most informative patterns in the data (the support vectors), hence interpretation is straightforward and data cleaning [8] could be implemented to improve performance. SVMs are the most well known of a class of algorithms which use the idea of kernel substitution and which we will broadly refer to as *kernel methods*.

In this tutorial we introduce this subject, describing the application of kernel methods to classification, regression and novelty detection and the different optimisation techniques that may be required during training. This tutorial is not exhaustive and many approaches (e.g. kernel PCA [30], density estimation [47], etc) have not been considered. More thorough treatments are contained in the books by Cristianini and Shawe-Taylor [6], Vapnik's classic textbook on statistical learning theory [42] and recent edited volumes [29,35].

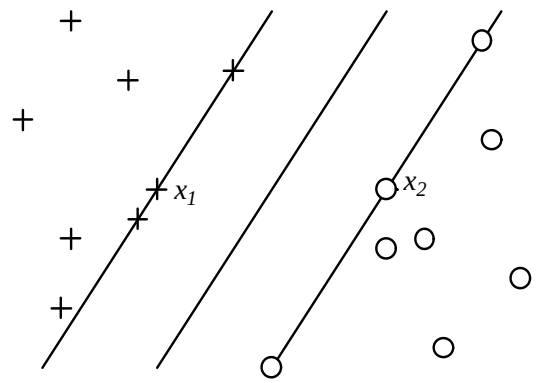

Fig. 1. The perpendicular distance between the separating hyperplane and a hyperplane through the closest points (the support vectors) is called the *margin*. $\mathbf{x}_1$ and $\mathbf{x}_2$ are examples of *support vectors* of opposite sign.

2 Learning with Support Vectors.

To introduce the subject we will begin by outlining the application of Support Vector Machines to the simplest case of binary classification. From the perspective of statistical learning theory the motivation for considering binary classifier SVMs comes from theoretical bounds on the generalisation error [42,6]. Though we do not quote the relevant theorem here we note that it has two important features. Firstly, the upper bound on the generalization error does not depend on the dimension of the space. Secondly, the error bound is minimised by maximising the *margin,* γ, i.e. the minimal distance between the hyperplane separating the two classes and the closest datapoints to the hyperplane (Figure 1).

Let us consider a binary classification task with datapoints $\mathbf{x}_i$ $(i = 1, \ldots, m)$ having corresponding labels $y_i = \pm 1$ and let the decision function be:

$$f(\mathbf{x}) = \text{sign}\,(\mathbf{w} \cdot \mathbf{x} + b) \tag{1}$$

If the dataset is separable then the data will be correctly classified if $y_i(\mathbf{w} \cdot \mathbf{x}_i + b) > 0 \ \forall i$. Clearly this relation is invariant under a positive rescaling of the argument inside the *sign*-function, hence we can define a *canonical hyperplane* such that $\mathbf{w} \cdot \mathbf{x} + b = 1$ for the closest points on one side and $\mathbf{w} \cdot \mathbf{x} + b = -1$ for the closest on the other side. For the separating hyperplane $\mathbf{w} \cdot \mathbf{x} + b = 0$ the normal vector is clearly $\mathbf{w}/\left|\left|\mathbf{w}\right|\right|_2$, and hence the margin is given by the projection of $\mathbf{x}_1 - \mathbf{x}_2$ onto this vector (see Figure 1). Since $\mathbf{w} \cdot \mathbf{x}_1 + b = 1$ and $\mathbf{w} \cdot \mathbf{x}_2 + b = -1$ this means the margin is $\gamma = 1/\left|\left|\mathbf{w}\right|\right|_2$. To maximise the margin the task is therefore:

$$\text{Minimise} \quad g(\mathbf{w}) = \frac{1}{2}\left|\left|\mathbf{w}\right|\right|_2^2 \tag{2}$$

subject to the constraints:

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) \geq 1 \qquad \forall i \tag{3}$$

and the learning task can be reduced to minimisation of the primal lagrangian:

$$L = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) - \sum_{i=1}^{m} \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x} + b) - 1 \right) \tag{4}$$

where α_i are Lagrangian multipliers (hence $\alpha_i \geq 0$). Taking the derivatives with respect to b and $\mathbf{w}$ and resubstituting back in the primal gives the Wolfe dual lagrangian:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j \right) \tag{5}$$

which must be maximised with respect to the α_i subject to the constraint:

$$\alpha_i \geq 0 \qquad \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{6}$$

So far we haven't used the second feature implied by the generalisation theorem mentioned above: the bound does not depend on the dimensionality of the space. For the dual lagrangian (5) we notice that the datapoints, $\mathbf{x}_i$, only appear inside an inner product. To get a better representation of the data we can therefore map the datapoints into an alternative higher dimensional space, called *feature space*, through a replacement:

$$\mathbf{x}_i \cdot \mathbf{x}_j \rightarrow \phi \left(\mathbf{x}_i \right) \cdot \phi(\mathbf{x}_j) \tag{7}$$

The functional form of the mapping $\phi(\mathbf{x}_i)$ does not need to be known since it is implicitly defined by the choice of *kernel:* $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ or inner product in feature space (feature space must therefore be a pre-Hilbert or inner product space). With a suitable choice of kernel the data can become separable in feature space despite being non-separable in the original input space: hence kernel substitution provides a route for obtaining nonlinear algorithms from algorithms previously restricted to handling linearly separable datasets. Thus, for example, whereas data for n-parity or the two spirals problem is non-separable by a hyperplane in input space it can be separated in the feature space defined by RBF kernels (giving an RBF-type network):

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-(\mathbf{x}_i - \mathbf{x}_j)^2 / 2\sigma^2} \tag{8}$$

Many other choices for the kernel function are possible e.g.:

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d \qquad K(\mathbf{x}_i, \mathbf{x}_j) = \tanh(\beta \mathbf{x}_i \cdot \mathbf{x}_j + b) \quad (9)$$

defining polynomial and feedforward neural network classifiers. Indeed, the class of mathematical objects which can be used as kernels is very general, and includes scores produced by dynamic alignment algorithms [10,45], for example. Suitable kernels must satisfy a mathematical condition: Mercer's theorem [19].

For binary classification with the given choice of kernel the learning task therefore involves maximisation of the lagrangian:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \quad (10)$$

subject to the constraints (6). The associated *Karush-Kuhn-Tucker* (KKT) conditions are:

$$\begin{aligned} y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 &\geq 0 \qquad \forall i \\ \alpha_i &\geq 0 \qquad \forall i \\ \alpha_i \left(y_i(\mathbf{w} \cdot \mathbf{x}_i + b) - 1\right) &= 0 \qquad \forall i \end{aligned} \quad (11)$$

which are always satisfied when a solution is found. After the optimal values of α_i have been found the decision function is based on the sign of:

$$f(\mathbf{z}) = \sum_{i=1}^{m} y_i \alpha_i K(\mathbf{z}, \mathbf{x}_i) + b \quad (12)$$

Since the bias, b, does not feature in the above dual formulation it is found from the primal constraints:

$$b = -\frac{1}{2} \left[\max_{\{i|y_i=-1\}} \left(\sum_{j \in \{\mathrm{SV}\}}^{m} y_j \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right) + \min_{\{i|y_i=+1\}} \left(\sum_{j \in \{\mathrm{SV}\}}^{m} y_j \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \right) \right] \quad (13)$$

We will henceforth refer to such a solution (α_i, b) as a *hypothesis* modelling the data.

When the maximal margin hyperplane is found in feature space, only those points which lie closest to the hyperplane have $\alpha_i > 0$ and these points are

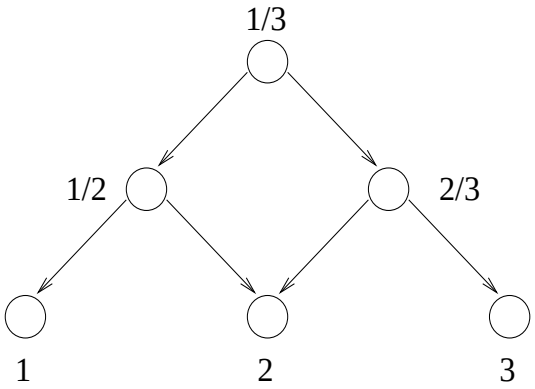

Fig. 2. A multi-class classification problem can be reduced to a series of binary classification tasks.

the *support vectors*. All other points have $\alpha_i = 0$. This means that the representation of hypothesis is solely given by those points which are closest to the hyperplane and *they are the most informative patterns in the data.*

Many problems involve multiclass classification and a number of schemes have been outlined [18,46] (with broadly similar performance). One of the simplest schemes is to use a directed graph (Figure 2) with the learning task reduced to binary classification at each node [25].

2.1 Soft margins and allowing for training errors.

Most real life datasets contain noise and an SVM can fit this noise leading to poor generalisation. The effect of outliers and noise can be reduced by introducing a *soft margin* [4] and two schemes are currently used. In the first (L_1 error norm) the learning task is the same as in (10,6) except for the introduction of the box constraint:

$$0 \leq \alpha_i \leq C \tag{14}$$

while in the second (L_2 error norm) the learning task is (10,6) except for addition of a small positive constant to the leading diagonal of the kernel matrix [4,26]:

$$K(\mathbf{x}_i, \mathbf{x}_i) \leftarrow K(\mathbf{x}_i, \mathbf{x}_i) + \lambda \tag{15}$$

C and λ control the trade-off between training error and generalisation ability and are chosen by means of a validation set. The effect of these soft margins is illustrated in Figure 3 for the ionosphere dataset from the UCI Repository [48].

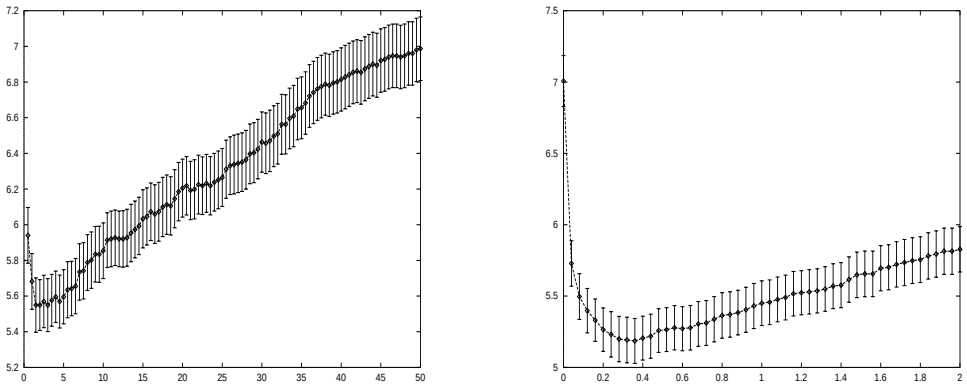

Fig. 3. *Left:* Generalisation error as a percentage (y-axis) versus C (x-axis) and *right:* generalisation error as a percentage (y-axis) versus λ (x-axis) for soft margin classifiers based on L_1 and L_2 error norms respectively. The UCI ionosphere dataset was used with RBF kernels ($\sigma = 1.5$) and 100 samplings of the data.

The justification for these soft margin techniques comes from statistical learning theory but can be readily viewed as relaxation of the *hard margin* constraint (3). Thus for the L_1 error norm (and prior to introducing kernels) we introduce a positive slack variable ξ_i into (3):

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) \geq 1 - \xi_i \tag{16}$$

and the task is now to minimise the sum of errors $\sum_{i=1}^{m} \xi_i$ in addition to $||\mathbf{w}||^2$:

$$\min \left[\frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i\right] \tag{17}$$

This is readily formulated as a primal objective function :

$$\begin{aligned} L(\mathbf{w}, b, \alpha, \xi) = & \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i \\ & - \sum_{i=1}^{m} \alpha_i \left[y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i \xi_i \end{aligned} \tag{18}$$

with Lagrange multipliers $\alpha_i \geq 0$ and $r_i \geq 0$. The derivatives with respect to $\mathbf{w}$, b and ξ give:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \tag{19}$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0 \tag{20}$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - r_i = 0 \tag{21}$$

Resubstituting these back in the primal objective function we obtain the same dual objective function, (10), as before. However, $r_i > 0$ and $C - \alpha_i - r_i = 0$, hence $\alpha_i \leq C$ and the constraint $0 \leq \alpha_i$ is replaced by $0 \leq \alpha_i \leq C$. Patterns with values $0 < \alpha_i < C$ will be referred to later as *non-bound* and those with $\alpha_i = 0$ or $\alpha_i = C$ will be said to be *at bound*. For an L_1 error norm we find the bias in the decision function (12) by using the final KKT condition in (11). Thus if i is a *non-bound* pattern it follows that $b = y_i - \sum_j \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j)$ assuming $y_i = \pm 1$.

The optimal value of C must be found by experimentation using a validation set and it cannot be readily related to the characteristics of the dataset or model. In an alternative approach (ν-SVM [33]) it can be shown that solutions for an L_1-error norm are the same as those obtained from maximising:

$$W(\alpha) = -\frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{22}$$

subject to:

$$\sum_{i=1}^{m} y_i \alpha_i = 0 \qquad \sum_{i=1}^{m} \alpha_i \geq \nu \qquad 0 \leq \alpha_i \leq \frac{1}{m} \tag{23}$$

where ν lies on the range 0 to 1. The fraction of training errors is upper bounded by ν and ν also provides a lower bound on the fraction of points which are support vectors. Hence in this formulation the conceptual meaning of the soft margin parameter is more transparent.

For the L_2 error norm the primal objective function is:

$$\begin{aligned} L(\mathbf{w}, b, \alpha, \xi) = & \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i^2 \\ & - \sum_{i=1}^{m} \alpha_i \left[y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) - 1 + \xi_i \right] - \sum_{i=1}^{m} r_i \xi_i \end{aligned} \tag{24}$$

with $\alpha_i \geq 0$ and $r_i \geq 0$. After obtaining the derivatives with respect to $\mathbf{w}$, b and ξ, substituting for $\mathbf{w}$ and ξ in the primal objective function and noting that the dual objective function is maximal when $r_i = 0$, we obtain the following

dual objective function after kernel substitution:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{4C} \sum_{i=1}^{m} \alpha_i^2 \tag{25}$$

With $\lambda = 1/2C$ this gives the same dual objective function as for hard margin learning except for the substitution (15). For many real-life datasets there is an imbalance between the amount of data in different classes, or the significance of the data in the two classes can be quite different. For example, for the detection of tumours on MRI scans it may be best to allow a higher number of false positives if this improved the true positive detection rate. The relative balance between the detection rate for different classes can be easily shifted [43] by introducing asymmetric soft margin parameters. Thus for binary classification with an L_1 error norm $0 \leq \alpha_i \leq C_+$ $(y_i = +1)$, and $0 \leq \alpha_i \leq C_-$ $(y_i = -1)$, while $K(\mathbf{x}_i, \mathbf{x}_i) \leftarrow K(\mathbf{x}_i, \mathbf{x}_i) + \lambda_+$ (if $y_i = +1$) and $K(\mathbf{x}_i, \mathbf{x}_i) \leftarrow K(\mathbf{x}_i, \mathbf{x}_i) + \lambda_-$ (if $y_i = -1$) for the L_2 error norm.

2.2 A Linear Programming Approach to Classification.

Rather than using quadratic programming it is also possible to derive a kernel classifier in which the learning task involves *linear programming* instead. Formulated directly in feature space this involves:

$$\min \left[\sum_{i=1}^{m} \alpha_i + C \sum_{i=1}^{m} \xi_i \right] \tag{26}$$

subject to:

$$y_i \left[\sum_{j=1}^{m} \alpha_i K(x_i, x_j) + b \right] \geq 1 - \xi_i \tag{27}$$

where $\alpha_i \geq 0$ and $\xi_i \geq 0$. By minimising $\sum_{i=1}^{m} \alpha_i$ we could obtain a solution which is *sparse* i.e. relatively fewer datapoints are used. Furthermore, efficient simplex or column generation implementations exist for solving linear programming problems so this is a practical alternative to conventional QP SVMs. This linear programming approach evolved independently of the QP approach to SVMs [17] and, as we will see, linear programming approaches to regression and novelty detection are also possible.

3 Novelty Detection.

For many real-world problems the task is not to classify but to detect novel or abnormal instances. Novelty or abnormality detection has potential applications in many problem domains such as condition monitoring or medical diagnosis. One approach is to model the *support* of a data distribution (rather than having to find a real-valued function for estimating the density of the data itself). Thus, at its simplest level, the objective is to create a binary-valued function which is positive in those regions of input space where the data predominantly lies and negative elsewhere.

One approach [36] is to find a hypersphere with a minimal radius R and centre $\mathbf{a}$ which contains most of the data: novel test points lie outside the boundary of this hypersphere. The technique we now outline was originally suggested by Vapnik [41,1], intepreted as a novelty detector by Tax and Duin [36] and used by the latter authors for real life applications [37]. The effect of outliers is reduced by using slack variables ξ_i to allow for datapoints outside the sphere and the task is to minimise the volume of the sphere and number of datapoints outside i.e.

$$\min \left[R^2 + \frac{1}{m\nu} \sum_i \xi_i \right] \tag{28}$$

subject to the constraints:

$$(\mathbf{x}_i - \mathbf{a})^T (\mathbf{x}_i - \mathbf{a}) \leq R^2 + \xi_i \tag{29}$$

and $\xi_i \geq 0$, and where ν controls the tradeoff between the two terms. The primal objective function is then:

$$\begin{aligned} L(R, \mathbf{a}, \alpha_i, \xi_i) = R^2 + \frac{1}{m\nu} \sum_{i=1}^{m} \xi_i - \sum_{i=1}^{m} \gamma_i \xi_i \\ - \sum_{i=1}^{m} \alpha_i \left(R^2 + \xi_i - (\mathbf{x}_i \cdot \mathbf{x}_i - 2\mathbf{a} \cdot \mathbf{x}_i + \mathbf{a} \cdot \mathbf{a}) \right) \end{aligned} \tag{30}$$

with $\alpha_i \geq 0$ and $\gamma_i \geq 0$. After kernel substitution the dual formulation amounts to maximisation of:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) - \sum_{i,j=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{31}$$

with respect to α_i and subject to $\sum_{i=1}^{m} \alpha_i = 1$ and $0 \leq \alpha_i \leq 1/m\nu$. If $m\nu > 1$ then *at bound* examples will occur with $\alpha_i = 1/m\nu$ and these correspond to outliers in the training process. Having completed the training process a test point $\mathbf{z}$ is declared novel if:

$$K(\mathbf{z},\mathbf{z}) - 2\sum_{i=1}^{m} \alpha_i K(\mathbf{z},\mathbf{x}_i) + \sum_{i,j=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_i,\mathbf{x}_j) - R^2 \geq 0 \tag{32}$$

where R^2 is first computed by finding an example which is *non-bound* and setting this inequality to an equality.

An alternative approach has been developed by Schölkopf et al. [31]. Suppose we restricted our attention to RBF kernels: in this case the data lie in a region on the surface of a hypersphere in feature space since $\phi(\mathbf{x})\cdot\phi(\mathbf{x}) = K(\mathbf{x},\mathbf{x}) = 1$ from (8). The objective is therefore to separate off this region from the surface region containing no data. This is achieved by constructing a hyperplane which is maximally distant from the origin with all datapoints lying on the opposite side from the origin and such that $\mathbf{w}\cdot\mathbf{x}_i + b \geq 0$. After kernel substitution the dual formulation of the learning task involves minimisation of:

$$W(\alpha) = \frac{1}{2}\sum_{i,k=1}^{m} \alpha_i \alpha_j K(\mathbf{x}_i,\mathbf{x}_j) \tag{33}$$

subject to:

$$0 \leq \alpha_i \leq \frac{1}{m\nu} \qquad \sum_{i=1}^{m} \alpha_i = 1 \tag{34}$$

To determine the bias we find an example, k say, which is non-bound (α_i and β_i are nonzero and $0 < \alpha_i < 1/m\nu$) and determine b from:

$$b = -\sum_{j=1}^{m} \alpha_j K(\mathbf{x}_j,\mathbf{x}_k) \tag{35}$$

The support of the distribution is then modelled by the decision function:

$$f(\mathbf{z}) = \text{sign}\left(\sum_{j=1}^{m} \alpha_j K(\mathbf{x}_j,\mathbf{z}) + b\right) \tag{36}$$

In the above models the parameter ν has a neat interpretation as an upper bound on the fraction of outliers and a lower bound of the fraction of patterns which are support vectors [31]. Schölkopf et al.[31] provide good experimental

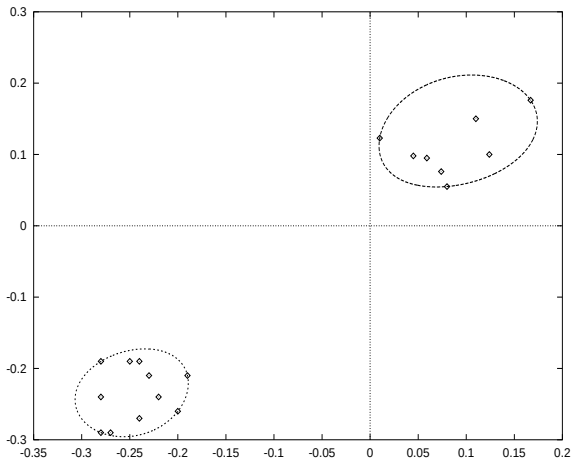

Fig. 4. The solution in input space for the hyperplane minimising $W(\alpha, b)$ in equation (37). A hard margin was used with RBF kernels trained using a $\sigma = 0.2$.

evidence in favour of this approach including the highlighting of abnormal digits in the USPS handwritten character dataset. The method also works well for other types of kernel. This and the earlier scheme for novelty detection can also be used with an L_2 error norm in which case the constraint $0 \leq \alpha_i \leq 1/m\nu$ is removed and an addition to the kernel diagonal is used instead.

For the model of Schölkopf et al. the origin of feature space plays a special role. It effectively acts as a prior for where the class of abnormal instances is assumed to lie. Rather than repelling away from the origin we could consider attracting the hyperplane onto datapoints in feature space. In input space this corresponds to a surface which wraps around the data clusters (Figure 4) and can be achieved through the following linear programming task [2]:

$$\min \left[\sum_{i=1}^{m} \left(\sum_{j=1}^{m} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b \right) \right] \tag{37}$$

subject to:

$$\sum_{j=1}^{m} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b \geq 0 \tag{38}$$

$$\sum_{i=1}^{m} \alpha_i = 1, \quad \alpha_i \geq 0 \tag{39}$$

The bias b is just treated as an additional parameter in the minimisation process, though unrestricted in sign. To handle noise and outliers we can

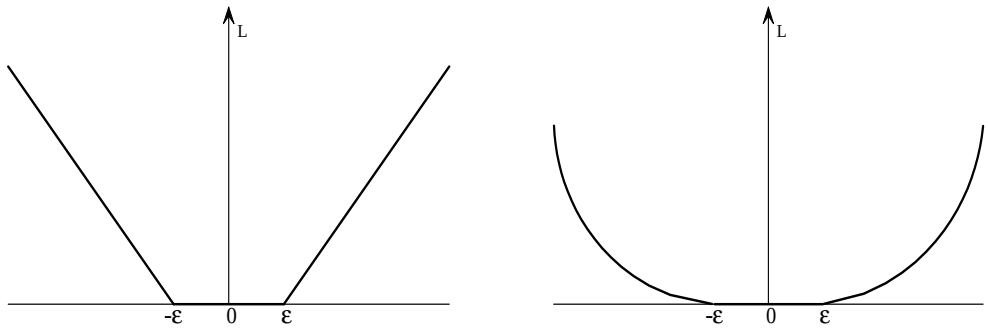

Fig. 5. Left figure: a linear ϵ-insensitive loss function versus $y_i - \mathbf{w} \cdot \mathbf{x}_i - b$. Right figure: a quadratic ϵ-insensitive loss function.

introduce a soft boundary:

$$\min \left[\sum_{i=1}^{m} \left(\sum_{j=1}^{m} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b \right) + \lambda \sum_{i=1}^{m} \xi_i \right] \tag{40}$$

subject to:

$$\sum_{j=1}^{m} \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + b \geq -\xi_i, \quad \xi_i \geq 0 \tag{41}$$

and constraints (39). This method has been successfully used for detection of abnormalities in blood samples and detection of faults in the condition monitoring of ball-bearing cages [2].

4 Regression.

For real-valued outputs the learning task can also be theoretically motivated from statistical learning theory. Instead of (3) we now use constraints $y_i - \mathbf{w} \cdot \mathbf{x}_i - b \leq \epsilon$ and $\mathbf{w} \cdot \mathbf{x}_i + b - y_i \leq \epsilon$ to allow for some deviation ϵ between the eventual targets y_i and the function $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, modelling the data. We can visualise this as a band or tube of size $\pm(\theta - \gamma)$ around the hypothesis function $f(\mathbf{x})$ and any points outside this tube can be viewed as training errors. The structure of the tube is defined by an $\epsilon-$*insensitive* loss function (Figure 5). As before we minimise $||\mathbf{w}||^2$ to penalise overcomplexity. To account for training errors we also introduce slack variables $\xi_i, \widehat{\xi}_i$ for the two types of training error. These slack variables are zero for points inside the tube and progressively increase for points outside the tube according to the loss function used. This general approach is called ϵ-SV regression [41] and is the most common approach to SV regression, though not the only one [42].

For a *linear $\epsilon-$insensitive loss function* the task is therefore to minimise:

$$\min \left[||\mathbf{w}||^2 + C \sum_{i=1}^{m} \left(\xi_i + \widehat{\xi}_i \right) \right] \tag{42}$$

subject to

$$\begin{aligned} y_i - \mathbf{w} \cdot \mathbf{x}_i - b &\leq \epsilon + \xi_i \\ (\mathbf{w} \cdot \mathbf{x}_i + b) - y_i &\leq \epsilon + \widehat{\xi}_i \end{aligned} \tag{43}$$

where the slack variables are both positive $\xi_i, \widehat{\xi}_i \geq 0$. After kernel substitution the dual objective function is:

$$\begin{aligned} W(\alpha, \widehat{\alpha}) = & \sum_{i=1}^{m} y_i(\alpha_i - \widehat{\alpha}_i) - \epsilon \sum_{i=1}^{m} (\alpha_i + \widehat{\alpha}_i) \\ & - \frac{1}{2} \sum_{i,j=1}^{m} (\alpha_i - \widehat{\alpha}_i)(\alpha_j - \widehat{\alpha}_j) K(x_i, x_j) \end{aligned} \tag{44}$$

which is maximised subject to

$$\sum_{i=1}^{m} \widehat{\alpha}_i = \sum_{i=1}^{m} \alpha_i \tag{45}$$

and:

$$0 \leq \alpha_i \leq C \qquad 0 \leq \widehat{\alpha}_i \leq C \tag{46}$$

Similarly a *quadratic $\epsilon-$insensitive loss function* gives rise to:

$$\min \left[||\mathbf{w}||^2 + C \sum_{i=1}^{m} \left(\xi_i^2 + \widehat{\xi}_i^2 \right) \right] \tag{47}$$

subject to (43), giving a dual objective function :

$$\begin{aligned} W(\alpha, \widehat{\alpha}) = & \sum_{i=1}^{m} y_i(\alpha_i - \widehat{\alpha}_i) - \epsilon \sum_{i=1}^{m} (\alpha_i + \widehat{\alpha}_i) \\ & - \frac{1}{2} \sum_{i,j=1}^{m} (\alpha_i - \widehat{\alpha}_i)(\alpha_j - \widehat{\alpha}_j) \left(K(\mathbf{x}_i, \mathbf{x}_j) + \delta_{ij}/C \right) \end{aligned} \tag{48}$$

which is maximised subject to (45). The function modelling the data is then:

$$f(\mathbf{z}) = \sum_{i=1}^{m} (\alpha_i - \widehat{\alpha}_i) K(\mathbf{x}_i, \mathbf{z}) + b \tag{49}$$

We still have to compute the bias, b, and we do so by considering the KKT conditions for regression. For a linear loss function prior to kernel substitution these are:

$$\begin{aligned} \alpha_i \left(\epsilon + \xi_i - y_i + \mathbf{w} \cdot \mathbf{x}_i + b\right) &= 0 \\ \widehat{\alpha}_i \left(\epsilon + \widehat{\xi}_i + y_i - \mathbf{w} \cdot \mathbf{x}_i - b\right) &= 0 \end{aligned} \tag{50}$$

where $\mathbf{w} = \sum_{j=1}^{m} y_j (\alpha_j - \widehat{\alpha}_j) \mathbf{x}_j$, and:

$$\begin{aligned} (C - \alpha_i)\, \xi_i &= 0 \\ (C - \widehat{\alpha}_i)\, \widehat{\xi}_i &= 0 \end{aligned} \tag{51}$$

From the latter conditions we see that only when $\alpha_i = C$ or $\widehat{\alpha}_i = C$ are the slack variables non-zero: these examples correspond to points outside the ϵ-insensitive tube. Hence from (50) we can find the bias from a non-bound example with $0 < \alpha_i < C$ using $b = y_i - \mathbf{w} \cdot \mathbf{x}_i - \epsilon$ and similarly for $0 < \widehat{\alpha}_i < C$ we can obtain it from $b = y_i - \mathbf{w} \cdot \mathbf{x}_i + \epsilon$. Though the bias can be obtained from one such example it is best to compute it using an average over all points on the margin.

Apart from the formulations given here it is possible to define other loss functions giving rise to different dual objective functions. In addition, rather than specifying ϵ *a priori* it is possible to specify an upper bound ν ($0 \leq \nu \leq 1$) on the fraction of points lying outside the band and then find ϵ by optimising over the primal objective function:

$$\frac{1}{2}||\mathbf{w}||^2 + C \left(\nu m \epsilon + \sum_{i=1}^{m} |y_i - f(\mathbf{x}_i)| \right) \tag{52}$$

with ϵ acting as an additional parameter to minimise over [28]. As for classification and novelty detection it is possible to formulate a linear programming approach to regression with [47]:

$$\min \left[\sum_{i=1}^{m} \alpha_i + \sum_{i=1}^{m} \alpha_i^* + C \sum_{i=1}^{m} \xi_i + C \sum_{i=1}^{m} \xi_i^* \right] \tag{53}$$

subject to:

$$y_i - \epsilon - \xi_i \leq \left(\sum_{j=1}^{m} \left(\alpha_j^\star - \alpha_j \right) K(x_i, x_j) \right) + b \leq y_i + \epsilon + \xi_i^\star \qquad (54)$$

Minimising the some of of the α_i approximately minimises the number of support vectors which favours sparse hypotheses with smooth functional approximations of the data. In this approach the kernel does not need to satisfy Mercer's condition [47].

5 Algorithmic Approaches

So far the methods we have considered have involved linear or quadratic programming. Linear programming can be implemented using column generation techniques [21] and many packages are available, e.g. CPLEX. For quadratic programming there are also many applicable techniques including quasi-Newton, conjugate gradient and primal-dual interior point methods [16]. Certain QP packages are readily applicable such as MINOS and LOQO. These methods can be used to train an SVM rapidly but they have the disadvantage that the kernel matrix is stored in memory. For small datasets this is practical and QP routines are the best choice, but for larger datasets alternative techniques have to be used. These split into two categories: techniques in which kernel components are evaluated and discarded during learning and *working set* methods in which an evolving subset of data is used. For the first category the most obvious approach is to sequentially update the α_i and this is the approach used by the Kernel Adatron (KA) algorithm [7]. For binary classification (with no soft margin or bias) this is a simple gradient ascent procedure on (10) in which $\alpha_i \geq 0$ initially and the α_i are subsequently sequentially updated using:

$$\alpha_i \leftarrow \beta_i \theta(\beta_i) \qquad \text{where} \qquad \beta_i = \alpha_i + \eta \left(1 - y_i \sum_{j=1}^{m} \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \right) \qquad (55)$$

and $\theta(\beta)$ is the Heaviside step function. The optimal learning rate η can be readily evaluated: $\eta = 1/K(\mathbf{x}_i, \mathbf{x}_i)$ and a sufficient condition for convergence is $0 < \eta K(\mathbf{x}_i, \mathbf{x}_i) < 2$. With the decision function (12) this method is very easy to implement and can give a quick impression of the performance of SVMs on classification tasks. It is equivalent to Hildreth's method in Optimisation theory and can be generalised to the case of soft margins and inclusion of a bias [16]. However, it is not as fast as most QP routines, especially on small datasets.

Chunking and Decomposition. Rather than sequentially updating the α_i the alternative is to update the α_i in parallel but using only a subset or *chunk* of data at each stage. Thus a QP routine is used to optimise the lagrangian on an initial arbitrary subset of data. The support vectors found are retained and all other datapoints (with $\alpha_i = 0$) discarded. A new working set of data is then derived from these support vectors and additional datapoints which maximally violate the storage constraints. This *chunking* process is then iterated until the margin is maximised. Of course, this procedure may still fail because the dataset is too large or the hypothesis modelling the data is not sparse (most of the α_i are non-zero, say). In this case *decomposition* [23] methods provide a better approach: these algorithms only use a fixed size subset of data with the α_i for the remainder kept fixed.

Decomposition and Sequential Minimal Optimisation (SMO). The limiting case of decomposition is the Sequential Minimal Optimisation (SMO) algorithm of Platt [24] in which only two α_i are optimised at each iteration. The smallest set of parameters which can be optimised with each iteration is plainly two if the constraint $\sum_{i=1}^{m} \alpha_i y_i = 0$ is to hold. Remarkably, if only two parameters are optimised and the rest kept fixed then it is possible to derive an analytical solution which can be executed using few numerical operations. The method therefore consists of a heuristic step for finding the best pair of parameters to optimise and use of an analytic expression to ensure the lagrangian increases monotonically. For the hard margin case the latter is easy to derive from the maximisation of δW with respect to the additive corrections a, b in $\alpha_i \to \alpha_i + a$ and $\alpha_j \to \alpha_j + b$, $(i \neq j)$. For the L_1 soft margin care must be taken to avoid violation of the constraints (14) leading to bounds on these corrections. The SMO algorithm has been refined to improve speed [15] and generalised to cover the above three tasks of classification [24], regression [34] and novelty detection [31]. Due to its decomposition of the learning task and speed it is probably the method of choice for training SVMs.

Model Selection. Apart from the choice of kernel the other indeterminate is the choice of the kernel parameter (e.g. σ in (8)). The kernel parameter can be found using cross-validation if sufficient data is available. However, recent model selection strategies can give a reasonable estimate for the kernel parameter without use of additional validation data. As a first attempt we can use a theorem stating that the generalisation error bound is reduced as the margin γ is increased. This theorem gives the upper bound as $R^2/m\gamma^2$ where R is the radius of the smallest ball containing the training data. At an optimum of (10) it is possible to show that $\gamma^2 = 1/\sum_i \alpha_i^0$ (where α_i^0 are the values of α_i at the optimum). Also for RBF kernels $R \simeq 1$ (the data lies on the surface of hypersphere since $\phi(\mathbf{x}) \cdot \phi(\mathbf{x}) = K(\mathbf{x}, \mathbf{x}) = 1$ from (8)) so the bound can be written $\sum_{i=1}^{m} \alpha_i^0/m$. Hence an estimate for σ can be found by sequentially training SVMs on the same dataset at successively larger values of σ, evaluating the bound from the α_i^0 for each case and choosing that value

of σ for which the bound is minimised. This method [5] will give a reasonable estimate if the data is spread evenly over the surface of the hypersphere but it is poor if the data lie in a flat ellipsoid, for example, since the radius R would be influenced by the largest deviations. More refined estimates therefore take into account the distribution of the data.

One approach [3] is to theoretically rescale data in feature space to compensate for uneven distributions. A more complex strategy along these lines has also been proposed by Schölkopf et al [32] which leads to an algorithm which has performed well in practice for a small number of datasets. The most economical way to use the training data is to use a *leave-one-out* procedure [3,13]. As an example, we consider a recent scheme proposed by Joachims [14]. In this approach the number of leave-one-out errors of an L_1-norm soft margin SVM is bounded by $|\{i : (2\alpha_i B^2 + \xi_i) \geq 1\}|/m$ where α_i are the solutions of the optimisation task in (10,6) and B^2 is an upper bound on $K(x_i, x_i)$ with $K(x_i, x_j) \geq 0$ (we can determine ξ_i from $y_i(\sum_j \alpha_j K(x_j, x_i) + b) \geq 1 - \xi_i$). Thus, for a given value of the kernel parameter, the leave-one-out error is estimated from this quantity (the system is *not* retrained with datapoints left out: the bound is determined using the α_i^0 from the solution of (10,6)). The kernel parameter is then incremented or decremented in the direction needed to lower the bound. This method has worked wekk on classification of text [14].

6 Further techniques based on kernel representations.

So far we have considered methods based on linear and quadratic programming. Here we shall consider further approaches which may utilise general nonlinear programming or other techniques. In particular, we will consider approaches to two issues: how to improve generalisation performance over standard SVMs and how to create hypotheses which are sparse.

For the dual of input space datapoints become hyperplanes and separating hyperplane becomes points. *Version space* is the space of all hypotheses (points) consistent with the data and this space is bounded by the set of hyperplanes representing the data. An SVM solution can be viewed as the centre of the largest inscribable hypersphere in version space: the support vectors correspond to those examples with hyperplanes tangentially touching this hyperphere (Figure 6). If version space is aspherical then the centre of the largest inscribed hypersphere does not appear to be the best choice. Indeed, a better choice would be the Bayes point: approximately the centre of mass of version space. Bayes Point Machines (BPMs) construct a hypothesis based on this centre of version space and this choice can be justified by theoretical arguments [22,44,27] in addition to having a geometric appeal.

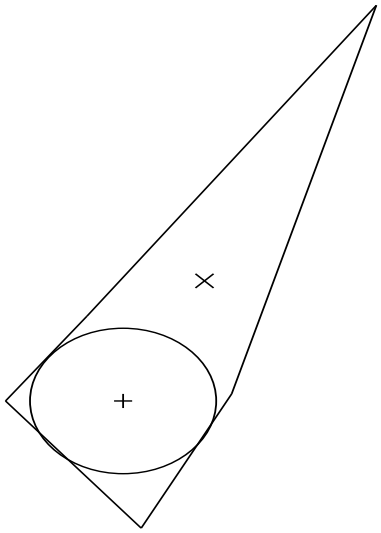

Fig. 6. The centre of mass of version space (+) and the centre of the largest inscribed sphere ($\times$) in an elongated version space

In one approach the centre of mass is determined using a kernelised billiard algorithm in which version space is traversed uniformly and an estimate of the centre of mass is repeatedly updated. For a large majority of datasets version space diverges from sphericality and the BPM outperforms an SVM at statistically significant levels. For artificial examples with very elongated version spaces the generalisation error of a BPM can be half that of an SVM [11,12]. However, current implementations of BPMs have a number of drawbacks: the algorithm can be slow in execution and better mechanisms for a soft boundary (imitating a soft margin) need to be found (the implementation in [12,11] also did not include a bias).

Rather than using the centre of mass of version space an alternative might be to use a hypothesis that lies towards the centre of this space but which is easier to compute. This could be achieved by using repulsive potentials $\Phi(\alpha)$ favouring points towards the centre of version space [27]. As an example we could use:

$$\min \Phi(\alpha) = \left[\sum_{i=1}^{m} \ln \left(\alpha_i K(x_i, x_j) + b \right) \right] \tag{56}$$

subject to:

$$\frac{1}{2} \sum_{i=1}^{m} \alpha_i^2 = 1 \tag{57}$$

which is the basis of the *Analytic Center Machine* [39]. The gradient and Hessian for (56) can be readily evaluated and the algorithm appears to perform

well in practice achieving a test error of 6.83% on a dataset for which an SVM gave 11.82%, for example.

The Bayes Point Machine may exhibit good generalisation but it has the disadvantage that the hypothesis is *dense* i.e. nearly all datapoints have $\alpha_i \neq 0$ and hence they appear in the final hypothesis. Ideally we would also like to derive kernel classifiers or regression machines which give *sparse* hypotheses using a minimal number of datapoints. The most effective means of obtaining sparse hypotheses remains the object of research but an excellent scheme is the Relevance Vector Machine of Tipping [38]. Using the function $f(z) = \sum_{i=1}^{m} \alpha_i K(z, x_i) + b$ with weights α_i, b to model the data, a Bayesian prior is defined over these parameters favouring smooth functions. From Bayes rule a posterior over the weights can be obtained and thence a marginal likelihood or evidence. Iterative maximisation of this evidence suggests suitable kernel values for pruning, creating an eventual hypothesis which is sparse in the number of datapoints used. Experiments show that this approach can sometimes give hypotheses which only use a few percent of the available data [38].

7 Conclusion

The approach we have considered is very general in that it can be applied to a wide range of machine learning tasks and can be used to generate many possible learning machine architectures (RBF networks, feedforward neural networks) through an appropriate choice of kernel. Above all, kernel methods have been found to work well in practice. The subject is still very much under development but it can be expected to develop as an important tool for machine learning and applications.

References

[1] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2, p. 121-167, 1998.

[2] C. Campbell and K. P. Bennett. A Linear Programming Approach to Novelty Detection. To appear in *Advances in Neural Information Processing Systems*, Vol. 14 MIT Press, 2001.

[3] O. Chapelle and V. Vapnik. Model selection for support vector machines, to appear in *Advances in Neural Information Processing Systems* 12, ed. S.A. Solla, T.K. Leen and K.-R. Muller, MIT Press, 2000.

[4] C. Cortes and V. Vapnik. Support vector networks. *Machine Learning* 20, p. 273-297, 1995.

[5] N. Cristianini, C. Campbell and J. Shawe-Taylor. Dynamically adapting kernels in support vector machines, *Advances in Neural Information Processing Systems* 11, ed. M. Kearns, S. A. Solla, and D. Cohn, MIT Press, p. 204-210, 1999.

[6] N. Cristianini and J. Shawe-Taylor. *An introduction to Support Vector Machines and other Kernel-based Learning Methods*, Cambridge University Press, 2000.

[7] T.-T. Friess, N. Cristianini and C. Campbell. The kernel adatron algorithm: a fast and simple learning procedure for support vector machines. *15th Intl. Conf. Machine Learning*, Morgan Kaufman Publishers, p. 188-196, 1998.

[8] I. Guyon, N. Matic and V. Vapnik. Discovering informative patterns and data cleaning. In U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, *Advances in Knowledge Discovery and Data Mining*, MIT Press, p. 181–203, 1996.

[9] Cf: http://www.clopinet.com/isabelle/Projects/SVM/applist.html

[10] D. Haussler. Convolution Kernels on Discrete Structures, UC Santa Cruz Technical Report UCS-CRL-99-10, 1999.

[11] R. Herbrich, T. Graepel and C. Campbell. Bayesian learning in reproducing kernel Hilbert spaces, submitted to *Machine Learning,* 1999.

[12] R. Herbrich, Th. Graepel and C. Campbell. *Proceedings of ESANN2000* (D-Facto Publications, Belgium, 2000) p. 49-54.

[13] T. Jaakolla and D. Haussler. Probabilistic kernel regression models, in *Proceedings of the 1999 Conference on AI and Statistics*, 1999.

[14] T. Joachims, Estimating the Generalization Performance of an SVM efficiently. *Proceedings of the Seventeenth International Conference on Machine Learning.* Morgan Kaufmann, 2000. p. 431-438.

[15] S. Keerthi, S. Shevade, C. Bhattacharyya and K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. Tech Report, Dept. of CSA, Banglore, India, 1999.

[16] D. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, 1984.

[17] O.L. Mangasarian. Linear and Nonlinear Separation of patterns by linear programming. *Operations Research* 13, p. 444-452, 1965.

[18] E. Mayoraz and E. Alpaydin. Support Vector Machines for Multiclass Classification, *Proceedings of the International Workshop on Artifical Neural Networks (IWANN99)*, IDIAP Technical Report 98-06, 1999.

[19] J. Mercer. Functions of positive and negative type and their connection with the theory of intregral equations. *Philos. Trans. Roy. Soc. London*, A209, p. 415-446, 1909.

[20] S. Mika, G. Ratsch, J. Weston, B. Schölkopf and K.-R. Muller. Fisher Discriminant Analysis with Kernels. *Proceedings of IEEE Neural Networks for Signal Processing Workshop* 1999, 8 pages, 1999.

[21] S. Nash and A. Sofer. Linear and Nonlinear Programming. McGraw-Hill, New York, NY, 1996.

[22] M. Opper and D. Haussler. Generalisation performance of bayes optimal classification algorithm for learning a perceptron. *Physical Review Letters*, 66, p. 2677-2680, 1991.

[23] E. Osuna and F. Girosi. Reducing the Run-time Complexity in Support Vector Machines, in B. Schölkopf, C.Burges and A. Smola (ed.), *Advances in Kernel Methods: Support Vector Learning,* MIT press, Cambridge, MA, p. 271-284, 1999.

[24] J. Platt. Fast training of SVMs using sequential minimal optimisation. in B. Schölkopf, C.Burges and A. Smola (ed.), *Advances in Kernel Methods: Support Vector Learning*, MIT press, Cambridge, MA, p. 185-208, 1999.

[25] J.Platt, N. Cristianini and J. Shawe-Taylor. Large Margin DAGS for Multiclass Classification, *Advances in Neural Information Processing Systems,* 12 ed. S.A. Solla, T.K. Leen and K.-R. Muller, MIT Press, 2000.

[26] J. Shawe-Taylor and N. Cristianini. Margin distribution and soft margin. In A. Smola, P. Barlett, B. Schölkopf and C. Schuurmans (eds), *Advances in Large Margin Classifiers*, Chapter 2, MIT Press, 1999.

[27] J. Schietse. Towards Bayesian Learning for the Perceptron. PhD thesis, Faculteit Wetenschappen, Limburgs Universitair Centrum, Belgium, 1996.

[28] B. Schölkopf, P. Bartlett, A. Smola and R. Williamson. Support vector regression with automatic accuracy control. In L. Niklasson, M. Bóden and T. Ziemke, editors, *Proceedings of the 8th International Conference on Artificial Neural Networks*, Perspectives in Neural Computing, Berlin, Springer Verlag, 1998.

[29] B. Schölkopf, C. Burges and A. Smola. *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA. 1998.

[30] B. Schölkopf, A. Smola and K.-R. Muller. Kernel Principal Component Analysis. In B. Schölkopf, C. Burges and A. Smola. *Advances in Kernel Methods: Support Vector Machines*. MIT Press, Cambridge, MA. 1998, p. 327-352.

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

[32] B. Schölkopf, J. Shawe-Taylor, A. Smola and R. Williamson. Kernel-dependent support vector error bounds, *Ninth International Conference on Artificial Neural Networks*, IEE Conference Publications No. 470, p. 304 - 309, 1999.

[33] B. Schölkopf, A. Smola, R. C. Williamson, and P. L. Bartlett. New support vector algorithms. To appear in *Neural Computation*.

[34] A. Smola and B. Schölkopf. A tutorial on support vector regression. NeuroColt2 TR 1998-03, 1998.

[35] A. Smola, P. Barlett, B. Schölkopf and C. Schuurmans (eds), *Advances in Large Margin Classifiers*, MIT Press, 2001.

[36] D. Tax and R. Duin. Data domain description by Support Vectors, in *Proceedings of ESANN99*, ed. M Verleysen, D. Facto Press, Brussels, p. 251-256, 1999.

[37] D. Tax, A. Ypma, and R. Duin. Support vector data description applied to machine vibration analysis. In: M. Boasson, J. Kaandorp, J.Tonino, M. Vosselman (eds.), *Proc. 5th Annual Conference of the Advanced School for Computing and Imaging* (Heijen, NL, June 15-17), 1999, 398-405.

[38] M. Tipping. The Relevance Vector Machine. In Sara A Solla, Todd K Leen, and Klaus-Robert Müller, editors, Advances in Neural Information Processing Systems 12. Cambridge, Mass: MIT Press, 2000.

[39] T. Trafalis and A. Malyscheff. An Analytic Center Machine. *Machine Learning*, to appear.

[40] V. Vapnik and O. Chapelle. Bounds on error expectation for SVMs, submitted to *Neural Computation*, 1999

[41] V. Vapnik. *The Nature of Statistical Learning Theory,* Springer, N.Y., 1995.

[42] V. Vapnik. *Statistical Learning Theory.* Wiley, 1998.

[43] K. Veropoulos, C. Campbell and N. Cristianini. Controlling the sensitivity of support vector machines. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, Stockholm, Sweden, 1999.

[44] T. Watkin and A. Rau. the Statistical Mechanics of Learning a Rule. *Reviews of Modern Physics* 65, p. 499-556, 1993.

[45] C. Watkins. Dynamic Alignment Kernels, Technical Report, UL Royal Holloway, CSD-TR-98-11, 1999.

[46] J. Weston and C. Watkins. Multi-Class Support Vector Machines, in *Proceedings of ESANN99,* ed. M Verleysen, D. Facto Press, Brussels, p. 219-224, 1999.

[47] J. Weston, A. Gammerman, M. Stitson, V. Vapnik, V. Vovk and C. Watkins. Support Vector Density Estimation. In B. Schölkopf, C. Burges and A. Smola. *Advances in Kernel Methods: Support Vector Machines.* MIT Press, Cambridge, MA. 1998, p. 293-306.

[48] http://www.ics.uci.edu/~mlearn/MLRepository.html