Least Squares Support Vector Machines Downloaded from www.worldscientific.com
by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

Chapter 2
Support Vector Machines

In this Chapter we give a short overview on the formulations of standard Support Vector Machines as introduced by Vapnik. We discuss linear and nonlinear SVM classifiers with the separable and non-separable case as well as linear and nonlinear function estimation by SVMs based on the Vapnik ϵ-insensitive loss function. The extension from the linear to the nonlinear case is done by the so-called kernel trick. In a second part of this Chapter, we discuss a number of modifications and extensions related to this SVM methodology. For textbooks and overview material we refer to [35; 51; 73; 202; 206; 219; 223; 279; 281].

2.1 Maximal margin classification and linear SVMs

2.1.1 *Margin*

While the weight decay term is an important aspect for obtaining good generalization in the context of neural networks for regression, the margin plays a somewhat similar role in classification problems. This margin concept is a first important step towards understanding the formulation of support vector machines.

In Fig. 2.1 an illustrative example is given of a separable problem in a two-dimensional input space. One can see that there exist several separating hyperplanes that separate the data of the two classes (data depicted by 'x' and '+'). On the other hand, one can also define a unique separating hyperplane. In order to achieve this, Vapnik considered a rescaling of the problem such that the points closest to the hyperplane satisfy

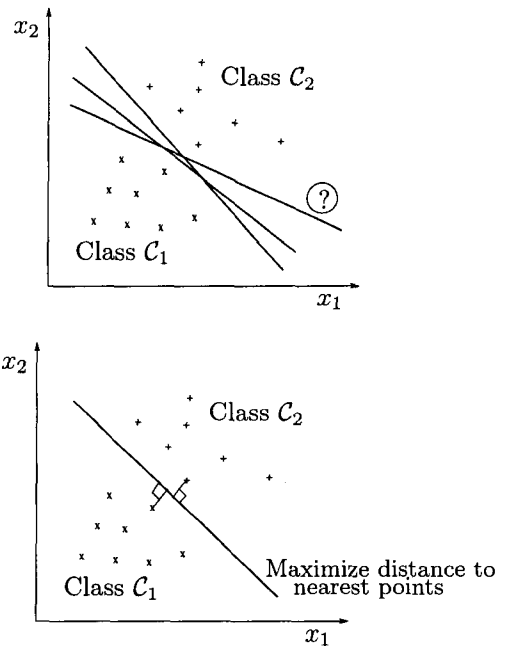


Fig. 2.1 *Linear classification: (Top) example of classification problem where the separating hyperplane is not unique; (Bottom) definition of a unique hyperplane which corresponds to a maximal distance between the nearest points of the two classes C_1 and C_2.*

$|w^T x_k + b| = 1$. One obtains a canonical form then for (w, b) of the hyperplane satisfying $y_k(w^T x_k + b) \geq 1$. In this case the margin equals $2/\|w\|_2$. Hence, the points closest to the hyperplane have a distance $1/\|w\|_2$ and in this way one maximizes the distance to the nearest points of the two classes. It is a desirable property then to maximize the margin which corresponds to minimizing $\|w\|_2$. This minimization of $w^T w$ is closely related to the use of a weight decay term in the training of neural networks.

2.1.2 *Linear SVM classifier: separable case*

Although the general nonlinear version of Support Vector Machines (SVM) is quite recent [279], the roots of the SVM approach about constructing an optimal separating hyperplane for pattern recognition dates back to the

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

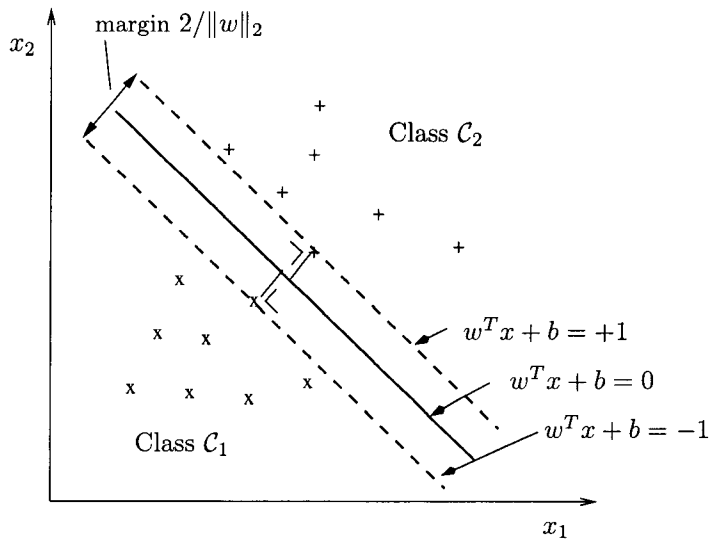


Fig. 2.2 *Linear classification: definition of a unique separating hyperplane illustrated in a two-dimensional input space. The margin is the distance between the dashed lines.*

work of Vapnik & Lerner in 1963 and Vapnik & Chervonenkis in 1964 [276; 277]. This original linear SVM formulation was made for separable data.

Consider a given training set $\{x_k, y_k\}_{k=1}^N$ with input data $x_k \in \mathbb{R}^n$ and output data $y_k \in \mathbb{R}$ with class labels $y_k \in \{-1, +1\}$ and the linear classifier

$$y(x) = \text{sign}[w^T x + b]. \tag{2.1}$$

When the data of the two classes are separable one can say

$$\begin{cases} w^T x_k + b \geq +1, & \text{if } y_k = +1 \\ w^T x_k + b \leq -1, & \text{if } y_k = -1. \end{cases} \tag{2.2}$$

These two sets of inequalities can be combined into one single set as follows

$$y_k[w^T x_k + b] \geq 1, \quad k = 1, ..., N. \tag{2.3}$$

Support vector machine formulations are done within a context of convex optimization theory. The general methodology is to start formulating the problem in the primal weight space as a constrained optimization problem, next formulate the Lagrangian, then take the conditions for optimality and finally solve the problem in the dual space of Lagrange multipliers. The latter will be called support values.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

One formulates an optimization problem which expresses that one should maximize the margin subject to the fact that all training data points need to be correctly classified. This gives the following primal problem in w:

$$\left[\begin{array}{ll} \boxed{\mathrm{P}}: & \min\limits_{w,b} J_{\mathrm{P}}(w) = \frac{1}{2} w^T w \\ & \text{such that} \quad y_k[w^T x_k + b] \geq 1, \quad k = 1, ..., N. \end{array}\right] \tag{2.4}$$

The Lagrangian for this problem is

$$\mathcal{L}(w, b; \alpha) = \frac{1}{2} w^T w - \sum_{k=1}^{N} \alpha_k \left(y_k [w^T x_k + b] - 1 \right) \tag{2.5}$$

with Lagrange multipliers $\alpha_k \geq 0$ for $k = 1, ..., N$. The solution is characterized by the saddle point of the Lagrangian

$$\max_{\alpha} \min_{w,b} \mathcal{L}(w, b; \alpha). \tag{2.6}$$

One obtains

$$\left\{\begin{array}{lll} \frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow & w = \sum\limits_{k=1}^{N} \alpha_k y_k x_k \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow & \sum\limits_{k=1}^{N} \alpha_k y_k = 0 \end{array}\right. \tag{2.7}$$

with resulting classifier

$$y(x) = \mathrm{sign}[\sum_{k=1}^{N} \alpha_k y_k x_k^T x + b]. \tag{2.8}$$

By replacing the expression for w (2.7) in the Lagrangian (2.5) one obtains the following Quadratic Programming (QP) problem as the dual problem in the Lagrange multipliers α_k

$$\left[\begin{array}{ll} \boxed{\mathrm{D}}: & \max\limits_{\alpha} J_{\mathrm{D}}(\alpha) = -\frac{1}{2} \sum\limits_{k,l=1}^{N} y_k y_l x_k^T x_l \ \alpha_k \alpha_l + \sum\limits_{k=1}^{N} \alpha_k \\ & \text{such that} \quad \sum\limits_{k=1}^{N} \alpha_k y_k = 0, \ \alpha_k \geq 0, \ \forall k \end{array}\right] \tag{2.9}$$

Note that this problem is solved in $\alpha = [\alpha_1; ...; \alpha_N]$, not in w. This QP problem has a number of interesting properties:

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

- *Global and unique solution*:
  The matrix related to the quadratic term in α in this quadratic form is positive definite or positive semidefinite. In the case that the matrix is positive definite (all eigenvalues strictly positive) the solution α to this QP problem is global and unique. When the matrix is positive semidefinite (all eigenvalues positive but zero eigenvalues possible) then the solution is global but not necessarily unique [82]. Some interesting discussions in relation to these aspects are given in [35]. It may happen that the solution of (w, b) is unique but α not. In terms of $w = \sum_k \alpha_k y_k x_k$ this means that there may exist equivalent expansions of w which require fewer support vectors.

- *Sparseness*:
  An interesting property is that many of the resulting α_k values are equal to zero. Hence the obtained solution vector is sparse. This means that in the resulting classifier the sum should be taken only over the non-zero α_k values instead of all training data points:

$$y(x) = \text{sign}[\sum_{k=1}^{\#\text{SV}} \alpha_k y_k x_k^T \, x + b]$$

  where the index k runs now over the number of support vectors. The training data points corresponding to non-zero α_k values are called *support vectors*.

- *Geometrical meaning of support vectors*:
  The support vectors obtained from the QP problem are located close to the decision boundary.

- *Non-parametric/parametric issues*:
  It is important to note that the size of the solution vector α grows with the number of data points N. Hence, the dual problem corresponds in fact to a non-parametric approach, while in the primal problem on the other hand the problem is parametric because the size of w is fixed independently of the number of data points. This means that for large data sets it might be advantageous to solve the primal problem but in higher dimensional input spaces it is better to solve the dual problem as the size of the solution vector α does

Least Squares Support Vector Machines Downloaded from www.worldscientific.com
by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

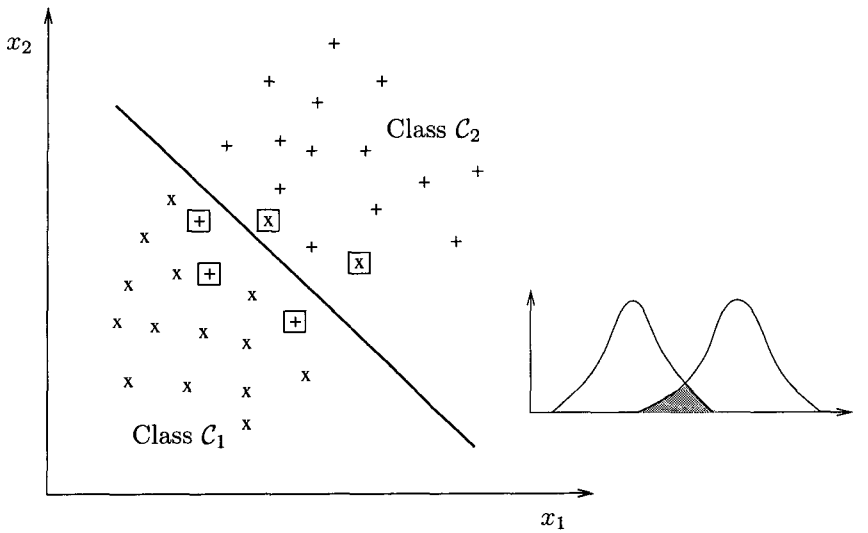


Fig. 2.3 *Problem of non-separable data, due to overlapping distributions.*

not depend on the dimension n of the input space.

2.1.3 *Linear SVM classifier: non-separable case*

In the previous subsection the SVM solution to a linearly separable classification problem was explained. On the other hand, one should note that most real-life problems that one encounters are *non-separable* cases either in a linear or nonlinear sense. Usually, one tries to find a set of inputs such that one can separate the classes as much as possible but often relevant inputs are missing, the data are incomplete, unreliable or noisy etc. Also in the previous Chapter it was shown that if for a binary classification problem the data are generated from a Gaussian density with the same covariance matrix for the two classes, the optimal decision boundary is a linear separating hyperplane according to Bayesian decision theory, no matter how small or large the overlap between these Gaussian densities. As a consequence this means that in the linearly non-separable case with overlapping distributions between the two classes, one should *tolerate misclassifications.*

The extension of linear SVMs to the non-separable case was made by

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

Cortes & Vapnik in 1995 [46]. Basically, it is done by taking additional slack variables in the problem formulation. In order to tolerate misclassifications, one modifies the set of inequalities into

$$y_k[w^T x_k + b] \geq 1 - \xi_k, \qquad k = 1, ..., N \tag{2.10}$$

with slack variables $\xi_k > 0$. When $\xi_k > 1$ the k-th inequality becomes violated in comparison with the corresponding inequality from the linearly separable case.

In the primal weight space the optimization problem becomes

$$\left[\boxed{\text{P}}: \quad \min_{w,b,\xi} J_{\text{P}}(w,\xi) = \frac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k \quad \text{such that} \quad y_k[w^T x_k + b] \geq 1 - \xi_k, \quad k = 1, ..., N \quad \xi_k \geq 0, \quad k = 1, ..., N \right] \tag{2.11}$$

where c is a positive real constant. The following Lagrangian should be considered then

$$\mathcal{L}(w, b, \xi; \alpha, \nu) = J_{\text{P}}(w, \xi) - \sum_{k=1}^{N} \alpha_k \left(y_k[w^T x_k + b] - 1 + \xi_k \right) - \sum_{k=1}^{N} \nu_k \xi_k \tag{2.12}$$

with Lagrange multipliers $\alpha_k \geq 0$, $\nu_k \geq 0$ for $k = 1, ..., N$. The second set of Lagrange multipliers is needed due to the additional slack variables ξ_k. The solution is given by the saddle point of the Lagrangian:

$$\max_{\alpha,\nu} \min_{w,b,\xi} \mathcal{L}(w, b, \xi; \alpha, \nu). \tag{2.13}$$

One obtains

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \quad \rightarrow \quad w = \sum_{k=1}^{N} \alpha_k y_k x_k \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \quad \rightarrow \quad \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \frac{\partial \mathcal{L}}{\partial \xi_k} = 0 \quad \rightarrow \quad 0 \leq \alpha_k \leq c, \ k = 1, ..., N \end{cases} \tag{2.14}$$

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

which gives the following dual QP problem after replacing (2.14) in (2.12):

$$\left[\begin{array}{lll} \boxed{\mathrm{D}}: & \max\limits_{\alpha} J_{\mathrm{D}}(\alpha) = & -\frac{1}{2}\sum\limits_{k,l=1}^{N} y_k y_l\, x_k^T x_l\, \alpha_k \alpha_l + \sum\limits_{k=1}^{N} \alpha_k \\ & \text{such that} & \sum\limits_{k=1}^{N} \alpha_k y_k = 0 \\ & & 0 \leq \alpha_k \leq c,\ k = 1, ..., N. \end{array}\right] \tag{2.15}$$

In comparison with the linearly separable case (2.9) this problem has additional box constraints.

2.2 Kernel trick and Mercer condition

Important progress in SVM theory has been made thanks to the fact that the linear method has been extended to a nonlinear technique by Vapnik in 1995 [279; 281]. In order to achieve this, one maps the input data into a high dimensional feature space which can be infinite dimensional. A construction of the linear separating hyperplane is done then in this high dimensional feature space*, after a nonlinear mapping $\varphi(x)$ of the input data to the feature space (Fig. 2.4).

Surprisingly, no explicit construction of the nonlinear mapping $\varphi(x)$ is needed. This is motivated by the following result. For any symmetric, continuous function $K(x, z)$ satisfying Mercer's condition [160], there exists a Hilbert space $\mathcal{H}$, a map $\varphi : \mathbb{R}^n \to \mathcal{H}$ and positive numbers λ_i such that one can write [48]:

$$K(x, z) = \sum_{i=1}^{n_{\mathcal{H}}} \lambda_i \phi_i(x) \phi_i(z), \tag{2.16}$$

where $x, z \in \mathbb{R}^n$ and $n_{\mathcal{H}}$ is the dimension of $\mathcal{H}$ (which can be infinite dimensional). Mercer's condition requires that

$$\int K(x, z) g(x) g(z) dx dz \geq 0 \tag{2.17}$$

*From the viewpoint of neural networks it would be better in fact to call this a high-dimensional *hidden layer*, because in pattern recognition one frequently uses the term feature space with another meaning of input space. Nevertheless we will use the term feature space in the sequel because it is always used in this area.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

for any square integrable function $g(x)$. The integral is taken here over a compact subset of $\mathbb{R}^n$. One can write $K(x,z) = \sum_{i=1}^{n_\mathcal{H}} \sqrt{\lambda_i}\phi_i(x)\sqrt{\lambda_i}\phi_i(z)$ and define $\varphi_i(x) = \sqrt{\lambda_i}\phi_i(x)$ and $\varphi_i(z) = \sqrt{\lambda_i}\phi_i(z)$ such that the kernel function can be expressed as the inner product (often called the dot product)

$$K(x,z) = \varphi(x)^T\varphi(z). \tag{2.18}$$

Hence, having a positive semidefinite† kernel is a condition to guarantee that one may write (2.18). This also implies that the kernel K is separable. Such separable kernels have been used for example also for solving integral equations [58].

The application of (2.18) is often called the *kernel trick*. It enables us to work in huge dimensional feature spaces without actually having to do explicit computations in this space. Computations are done in another space after applying this kernel trick. In the case of support vector machines, one starts from a formulation in the primal weight space with a high dimensional feature space by applying transformations $\varphi(\cdot)$. The problem is not solved in this primal weight space but in the dual space of Lagrange multipliers after applying the kernel trick. In this way one can implicitly work in high dimensional feature spaces (hidden layer space) without doing computations in that space (Fig. 2.5).

2.3 Nonlinear SVM classifiers

The extension from linear SVM classifiers to nonlinear SVM classifiers is straightforward. One can in fact formally replace x by $\varphi(x)$ and apply the kernel trick where possible. One should, however, be aware that $\varphi(x)$ can be infinite dimensional, and hence, also the w vector. While for linear SVMs one can in fact equally well solve the primal problem in w as the dual problem in the support values α, this is no longer the same for the nonlinear SVM case because in the primal problem the unknown w can be infinite dimensional.

In a similar way as for the linear SVM case, we can now write for the

†We use here the linear algebra terminology of *positive definite* and *positive semidefinite*. The corresponding terminology in functional analysis is often *strictly positive definite* and *positive definite*, respectively.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

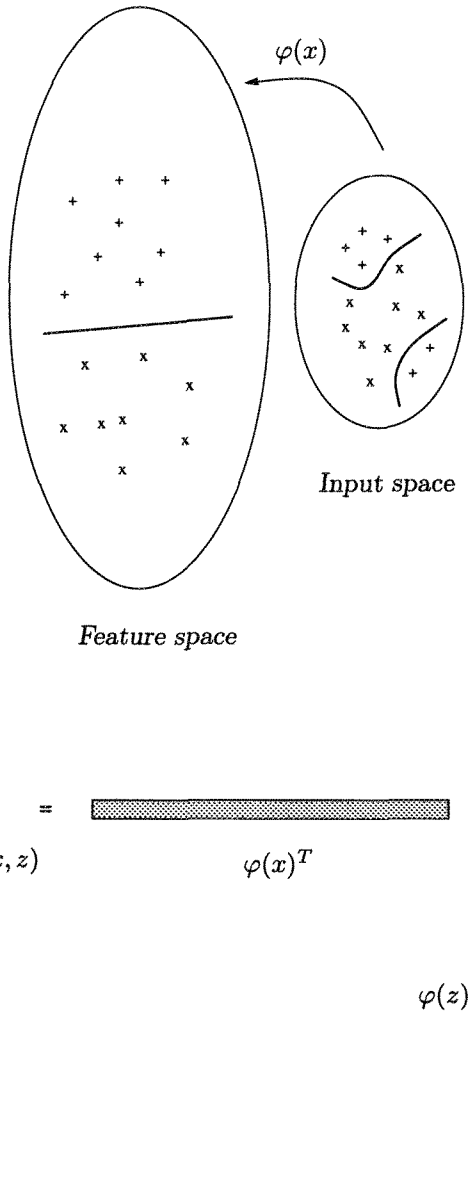


Fig. 2.4 *(Top) Mapping of the input space to a high dimensional feature space where a linear separation is made, which corresponds to a nonlinear separation in the original input space; (Bottom) Illustration of the kernel trick where $K(x,z)$ is chosen and the inner product representation exists provided $K(\cdot,\cdot)$ is a symmetric positive definite kernel.*

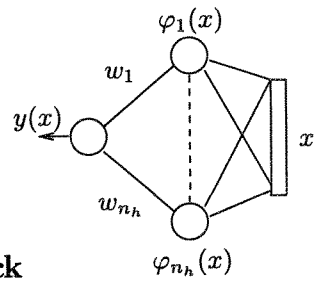
Least Squares Support Vector Machines Downloaded from www.worldscientific.com
by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com
by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

Primal problem P

Parametric: estimate $w \in \mathbb{R}^{n_h}$

$y(x) = \text{sign}[w^T \varphi(x) + b]$



Kernel trick

$K(x_k, x_l) = \varphi(x_k)^T \varphi(x_l)$

Dual problem D

Non-parametric: estimate $\alpha \in \mathbb{R}^N$

$y(x) = \text{sign}[\sum_{k=1}^{\#\text{sv}} \alpha_k y_k K(x, x_k) + b]$

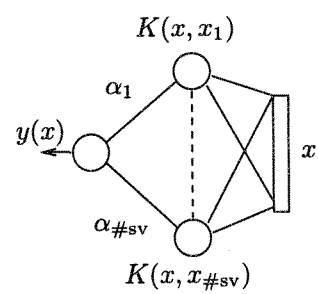


Fig. 2.5 *Primal-dual neural network interpretations of support vector machines.*

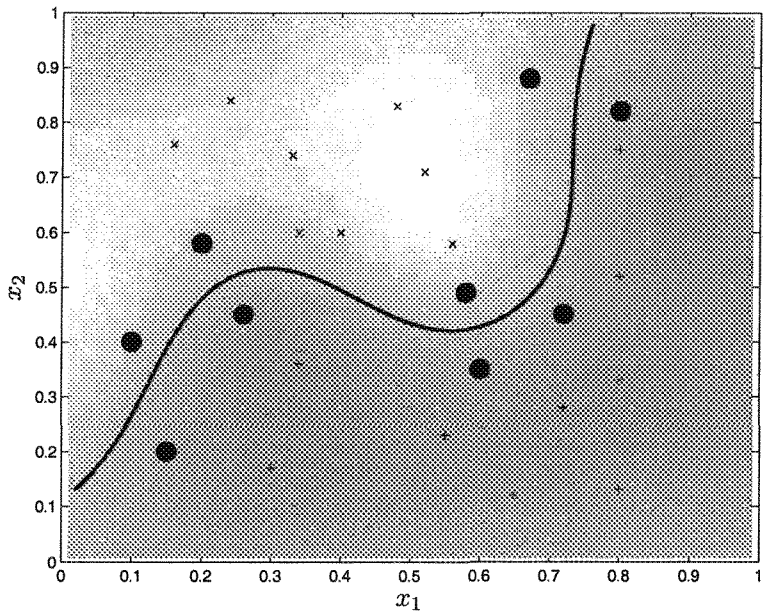


Fig. 2.6 *For nonlinear SVMs the support vectors (indicated by the black dots) are located close to the decision boundary as illustrated here by a simple nonlinearly separable binary class problem.*

nonlinear case

$$\begin{cases} w^T\varphi(x_k) + b \geq +1, & \text{if } y_k = +1 \\ w^T\varphi(x_k) + b \leq -1, & \text{if } y_k = -1 \end{cases} \tag{2.19}$$

which is equivalent to

$$y_k[w^T\varphi(x_k) + b] \geq 1, \quad k = 1, ..., N \tag{2.20}$$

in the case of separable data. At this point no explicit form or construction is made of $\varphi(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$. The number n_h is the number of hidden units in the primal weight space representation of the nonlinear classifier (Fig. 2.5)

$$y(x) = \text{sign}[w^T\varphi(x) + b] \tag{2.21}$$

where n_h corresponds to the dimension $n_{\mathcal{H}}$.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

The optimization problem becomes

$$\left[\begin{array}{lll} \boxed{\mathrm{P}}: \quad \min\limits_{w,b,\xi} J_{\mathrm{P}}(w,\xi) = & \frac{1}{2} w^T w + c \sum\limits_{k=1}^{N} \xi_k & \\ \text{such that} & y_k[w^T \varphi(x_k) + b] \geq 1 - \xi_k, & k = 1, ..., N \\ & \xi_k \geq 0, \quad k = 1, ..., N. & \end{array}\right] \tag{2.22}$$

One constructs the Lagrangian:

$$\mathcal{L}(w, b, \xi; \alpha, \nu) = J_{\mathrm{P}}(w, \xi) - \sum_{k=1}^{N} \alpha_k \left(y_k[w^T \varphi(x_k) + b] - 1 + \xi_k \right) - \sum_{k=1}^{N} \nu_k \xi_k \tag{2.23}$$

with Lagrange multipliers $\alpha_k \geq 0$, $\nu_k \geq 0$ for $k = 1, ..., N$. The solution is given by the saddle point of the Lagrangian:

$$\max_{\alpha,\nu} \min_{w,b,\xi} \mathcal{L}(w, b, \xi; \alpha, \nu). \tag{2.24}$$

One obtains

$$\left\{\begin{array}{lll} \frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow & w = \sum\limits_{k=1}^{N} \alpha_k y_k \varphi(x_k) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow & \sum\limits_{k=1}^{N} \alpha_k y_k = 0 \\ \frac{\partial \mathcal{L}}{\partial \xi_k} = 0 & \rightarrow & 0 \leq \alpha_k \leq c, \ k = 1, ..., N. \end{array}\right. \tag{2.25}$$

The quadratic programming problem (dual problem) becomes

$$\left[\begin{array}{lll} \boxed{\mathrm{D}}: \quad \max\limits_{\alpha} J_{\mathrm{D}}(\alpha) = & -\frac{1}{2} \sum\limits_{k,l=1}^{N} y_k y_l \, K(x_k, x_l) \, \alpha_k \alpha_l + \sum\limits_{k=1}^{N} \alpha_k & \\ \text{such that} & \sum\limits_{k=1}^{N} \alpha_k y_k = 0 & \\ & 0 \leq \alpha_k \leq c, \ k = 1, ..., N. & \end{array}\right] \tag{2.26}$$

In this quadratic form one makes use of the kernel trick

$$K(x_k, x_l) = \varphi(x_k)^T \varphi(x_l) \tag{2.27}$$

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

for $k, l = 1, ..., N$. Finally, the nonlinear SVM classifier takes the form

$$y(x) = \text{sign}[\sum_{k=1}^{N} \alpha_k \, y_k \, K(x, x_k) + b] \tag{2.28}$$

with α_k positive real constants which are the solution to the QP problem. However, we still need to determine b. The complementarity conditions from the Karush-Kuhn-Tucker (KKT) conditions (see Appendix) for the above problem state that the product of the dual variables and the constraints should be zero at the optimal solution. The KKT conditions yield

KKT :

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial w} = 0 \rightarrow & w = \sum_{k=1}^{N} \alpha_k y_k \varphi(x_k) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 \rightarrow & \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \frac{\partial \mathcal{L}}{\partial \xi_k} = 0 \rightarrow & c - \alpha_k - \nu_k = 0 \\ & \alpha_k \{ y_k [w^T \varphi(x_k) + b] - 1 + \xi_k \} = 0 \,, \quad k = 1, .., N \\ & \nu_k \xi_k = 0 \,, \quad k = 1, .., N \\ & \alpha_k \geq 0 \,, \quad k = 1, .., N \\ & \nu_k \geq 0 \,, \quad k = 1, .., N. \end{cases} \tag{2.29}$$

From $\nu_k \xi_k = 0$ we have for the solution $w^\star, b^\star, \xi^\star, \alpha^\star, \nu^\star$ to this problem that $\xi_k^\star = 0$ for $\alpha_k^\star \in (0, c)$. Hence

$$y_k [w^T \varphi(x_k) + b] - 1 = 0 \text{ for } \alpha_k \in (0, c), \tag{2.30}$$

which means that one can take any training data point for which $0 < \alpha_k < c$ and use that equation to compute the bias term b (numerically it might be better to take an average over these training data points).

Some properties of the nonlinear SVM classifier and its solution are the following:

- *Choice of kernel function:*
  Several choices are possible for the kernel $K(\cdot, \cdot)$. Some typical choices are

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

$$K(x, x_k) = x_k^T x \text{ (linear SVM)}$$
$$K(x, x_k) = (\tau + x_k^T x)^d \text{ (polynomial SVM of degree } d)$$
$$K(x, x_k) = \exp(-\|x - x_k\|_2^2/\sigma^2) \text{ (RBF kernel)}$$
$$K(x, x_k) = \tanh(\kappa_1 x_k^T x + \kappa_2) \text{ (MLP kernel) .}$$

The Mercer condition holds for all σ values in the RBF kernel case and positive τ values in the polynomial case but not for all possible choices of κ_1, κ_2 in the MLP kernel case. A further discussion about kernel functions will be given in the sequel of this Chapter.

- *Global and unique solution*:
  As in the linear SVM case the solution to the convex QP problem is again global and unique provided that one chooses a positive definite kernel for $K(\cdot, \cdot)$. This choice guarantees that the matrix involved in the QP problem is positive definite as well, and that one can apply the kernel trick. For a positive semidefinite kernel the solution to the QP problem is global but not necessarily unique.

- *Sparseness*:
  As in the linear SVM classifier case many α_k values are equal to zero in the solution vector. In the dual space the nonlinear SVM classifier takes the form

$$y(x) = \text{sign}[\sum_{k=1}^{\#\text{SV}} \alpha_k y_k K(x, x_k) + b]$$

  where the sum is taken over the non-zero α_k values which correspond to support vectors x_k of the training data set. In Fig. 2.5 a neural network interpretation is given of this. The number of hidden units equals the number of support vectors SV, that one finds from a convex QP problem. In the case of an RBF kernel, one has

$$\begin{aligned} y(x) &= \text{sign}[\sum_{k=1}^{N} \alpha_k y_k \exp\left(-\|x - x_k\|_2^2/\sigma^2\right) + b] \\ &= \text{sign}[\sum_{k=1}^{\#\text{SV}} \alpha_k y_k \exp\left(-\|x - x_k\|_2^2/\sigma^2\right) + b] \end{aligned} \tag{2.31}$$

  where # SV denotes the number of support vectors.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

- *Geometrical meaning of support vectors*:
  The support vectors obtained from the QP problem are located close to the nonlinear decision boundary illustrated on (Fig. 2.6) and (Fig. 2.7) for the synthetic Ripley data set [193].

- *Non-parametric/parametric issues*:
  Both the primal and the dual problem have neural network representation interpretations (Fig. 2.5). The problem in the primal weight space is parametric, while the dual problem is non-parametric. Note that in the dual problem the size of the QP problem is not influenced by the dimension n of the input space.

In comparison with traditional multilayer perceptron neural networks that suffer from the existence of multiple local minima solutions, convexity is an important and interesting property of nonlinear SVM classifiers. However, strictly speaking, this is less surprising if one notes that w and α are in fact interconnection weights of the output layer. Also in MLPs one would have a convex problem if one would fix a hidden layer interconnection matrix and one would compute the output layer (with linear characteristic at the output) from a sum squared error cost function. The nice aspect, however, of SVMs is the powerful representation of the network. If one takes an RBF kernel, almost all unknowns follow as the solution to a convex problem, which is not the case for traditional MLP networks. Furthermore, in classical MLPs one has to fix the number of hidden units beforehand while in nonlinear SVMs this number of hidden units follows from the QP problem as the number of support vectors.

2.4 VC theory and structural risk minimization

2.4.1 *Empirical risk versus generalization error*

From the linear and nonlinear SVM formulations it is clear so far that the margin plays an important role in the formulations. This importance can be put within the broader perspective of statistical learning theory (VC theory) [279; 281]. While in classical neural networks practice one selects models based upon specific validation sets, a major goal of VC (Vapnik-Chervonenkis) theory is to characterize the generalization error instead of the error on specific data sets.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

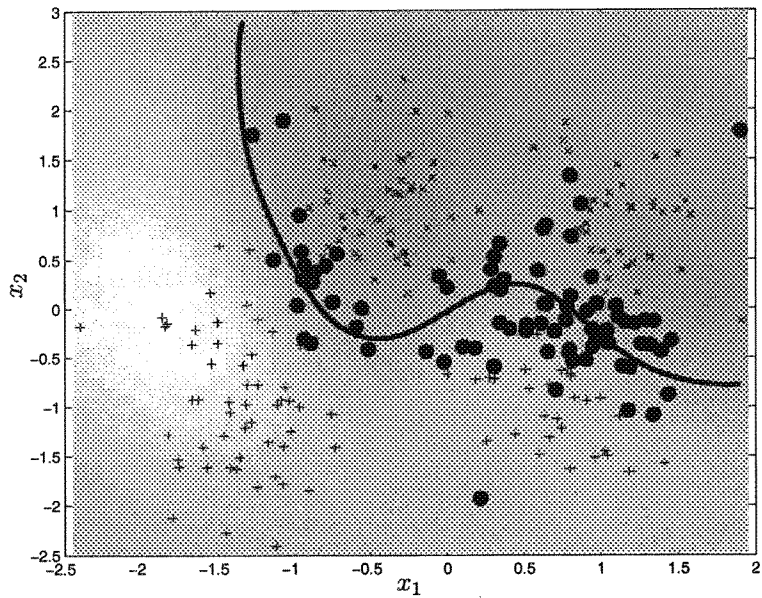


Fig. 2.7 *Nonlinear SVM classifier with RBF kernel on the Ripley data set binary classification problem with '+' and 'x' given training data points of the two classes. The black dots correspond to the support vectors, which have non-zero α_k values.*

Let us consider a binary classification problem and a set of functions with adjustable parameter vector θ such that $\{f(x;\theta) : \mathbb{R}^n \to \{-1,+1\}\}$ together with a training set $\{(x_k, y_k)\}_{k=1}^N$ with input patterns $x_k \in \mathbb{R}^n$ and output $y_k \in \{-1,+1\}$. The following error defined on the training data set is usually called empirical risk

$$R_{\text{emp}}(\theta) = \frac{1}{2N}\sum_{k=1}^{N}|y_k - f(x_k;\theta)|. \tag{2.32}$$

The generalization error (or risk) on the other hand is defined as

$$R(\theta) = \int \frac{1}{2}|y - f(x;\theta)|\, p(x,y)\, dxdy \tag{2.33}$$

which measures the error for all input/output patterns that are generated from the underlying generator of the data characterized by the probability

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

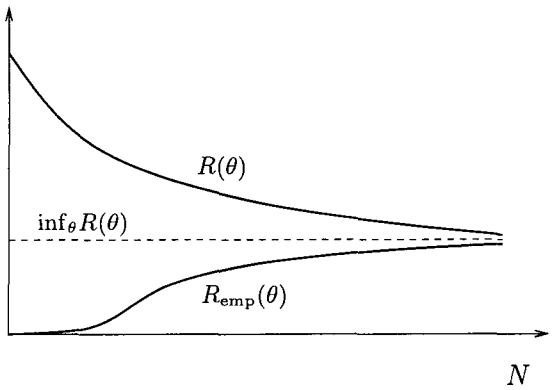


Fig. 2.8 *If the expected risk $R(\theta)$ and the empirical risk $R_{emp}(\theta)$ both converge to the value $\inf_\theta R(\theta)$ when the number of data N goes to infinity, then the learning process is consistent.*

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

distribution $p(x, y)$ (which is unfortunately not known in practice). However, one can derive bounds on this generalization error.

An important result by Vapnik (1979) states that one can give an upper bound to this generalization error in a probabilistic sense [279; 281]. Vapnik showed that for any parameter vector θ of the set of function $\{f(x;\theta)|\theta \in \Theta\}$, i.i.d. data and $N > h$, the bound

$$R(\theta) \leq R_{\text{emp}}(\theta) + \sqrt{\frac{h(\ln(2N/h) + 1) - \ln(\eta/4)}{N}} \tag{2.34}$$

holds with probability $1 - \eta$, where the second term is a confidence term which depends on the VC dimension h that characterizes the capacity of the set of functions and it is a combinatorial measure for the model complexity. An important aspect of this risk bound is that it assumes no specific form for the underlying density $p(x, y)$, only that i.i.d. data are drawn from this density $p(x, y)$. In addition to this risk upper bound also a lower bound on $R(\theta)$ can be derived. An important insight of VC theory is that the worst case over all functions that a learning machine can implement determines consistency (Fig. 2.8) of the empirical risk minimization. Without restricting the set of admissible functions, empirical risk minimization is not consistent. For the empirical risk minimization principle to be consistent, it is necessary and sufficient that the empirical risk $R_{\text{emp}}(\theta)$ converges uniformly to the risk $R(\theta)$ in the following sense: $\lim_{N\to\infty} P\{\sup_{\theta\in\Theta}(R(\theta) - R_{\text{emp}}(\theta)) > \epsilon\} = 0, \forall\epsilon > 0$ [279].

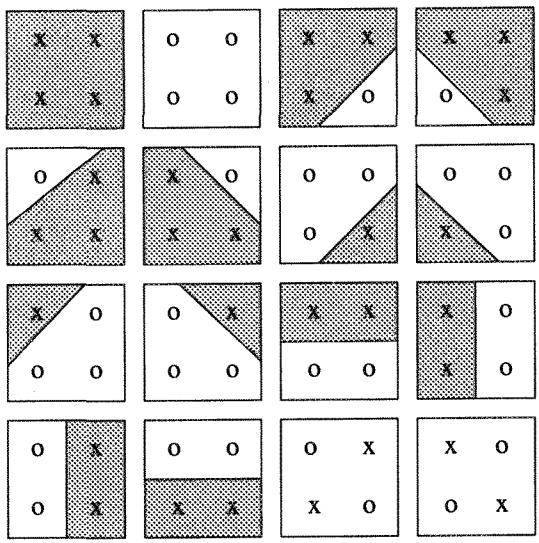


Fig. 2.9 *Example of $N = 4$ points in an $n = 2$ dimensional input space. The points can be labelled then in $2^4 = 16$ possible ways. At most 3 points can be separated by straight lines, because it's well known that the last two cases are XOR problems which cannot be separated by a straight line. This results in a VC dimension equal to 3. For nonlinear classifiers this VC dimension can be larger.*

The VC dimension can be understood as follows. If a given set of N points can be labelled in all possible 2^N ways and for each labelling a member of the set $\{f(x;\theta)\}$ can be found which correctly assigns those labels, we can say that that set of points is *shattered* (separated) by that set of functions. The VC dimension for the set of functions $\{f(x;\theta)\}$ is defined as the maximum number of training points that can be shattered by members of $\{f(x;\theta)\}$. For linear separating hyperplanes in an input space $\mathbb{R}^n$, one can show that the VC dimension equals $h = n + 1$. Consider for example $N = 4$ points in an $n = 2$ dimensional input space. The points can be labelled then in $2^4 = 16$ possible ways. Not all of these 16 cases are linearly separable, e.g., it is well known that XOR cases are not linearly separable (Fig. 2.9). At most 3 points can be separated or said in terms of the existing connection between VC theory and Popper's work in philosophy: four points can falsify any linear law for this example. In larger

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

dimensions one can relate this problem of linear separability also to Cover's theorem [23].

Intuitively one would expect that models with many parameters lead to a high VC dimension and models with few parameters to a low VC dimension. However, this intuition is not entirely true as it is easy to construct counterexamples of models having only one parameter that can possess an infinite VC dimension [35].

In the previous Section we have discussed the prominent role played by convex optimization within SVM methodology. On the other hand one should be aware of the fact that if one takes e.g. an RBF kernel, then the choice of σ for this RBF kernel does not follow as the solution to the QP problem. It should be fixed beforehand. In fact one has to try several possible values for this tuning parameter, in combination also with the other constant c in the algorithm. This VC upper bound (2.34) can then be used in order to select a good value of σ, c which guarantees a good generalization of the model. Although one exactly knows the VC dimension for linear classifiers, for nonlinear SVM classifiers no exact expressions are available for the VC dimension h. One often works then with an upper bound on h.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

2.4.2 *Structural risk minimization*

In order to apply the VC bound to SVM classifiers, Vapnik employs the concept of so-called structural risk minimization. One considers a structure

$$\mathcal{S}_i = \{w^T\varphi(x) + b : \|w\|_2^2 \leq c_i\}, \quad c_1 < c_2 < ... < c_i < c_{i+1} < ... \qquad (2.35)$$

which consists of nested sets of functions of increasing complexity. This is illustrated in Fig. 2.10 [206]. The concept is quite similar to previously discussed complexity criteria and bias-variance trade-off curves. In this case of VC theory one may consider sets of functions with increasing VC dimension. The larger this VC dimension the smaller the training set error can become (empirical risk) but the confidence term (second term in (2.34)) will grow. The minimum of the sum of these two terms is then a good compromise solution as the trade-off between these two curves.

For SVM classifiers one can find an upper bound on the VC dimension as follows. Vapnik has shown that hyperplanes satisfying $\|w\|_2 < a$ have a

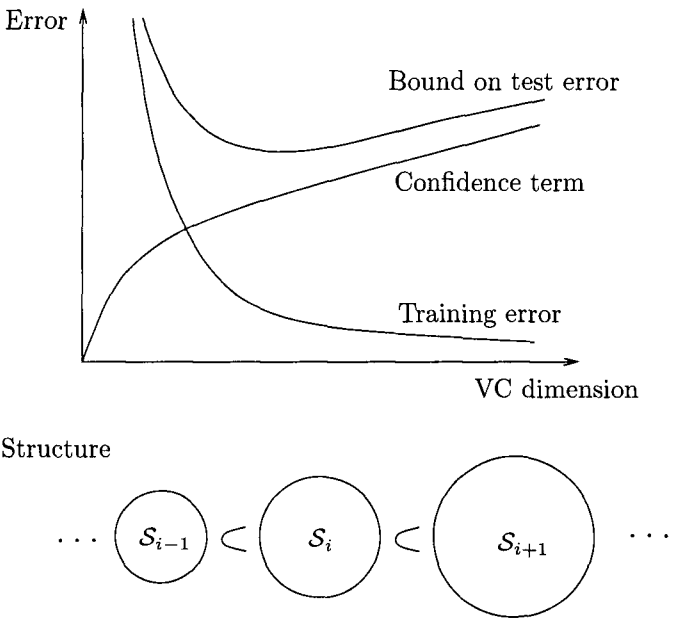


Fig. 2.10 *Principle of structural risk minimization and illustration of the generalization bound depending on the VC dimension.*

VC-dimension h which is upper bounded by

$$h \leq \min([r^2a^2], n) + 1 \tag{2.36}$$

where $[\cdot]$ denotes the integer part and r is the radius of the smallest ball containing the points $\varphi(x_1), ..., \varphi(x_N)$ in the high dimensional feature space (Fig. 2.11).

SVM parameters which do not result from the QP problem such as the σ value of the RBF kernel are often selected then in such a way that this upper bound (2.36) is minimal. A nice property is that this upper bound can again be computed by solving a convex QP problem. Hence one may proceed then by solving in an alternating way QP problems in α and QP problems related to (2.36), both by convex optimization. The value w^Tw can be computed by applying the kernel trick. One has $w^Tw = \alpha^T\Omega\alpha$ with $\Omega_{kl} = y_k y_l K(x_k, x_l)$ where $K(x_k, x_l) = \varphi(x_k)^T\varphi(x_l)$ for $k, l = 1, ..., N$.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

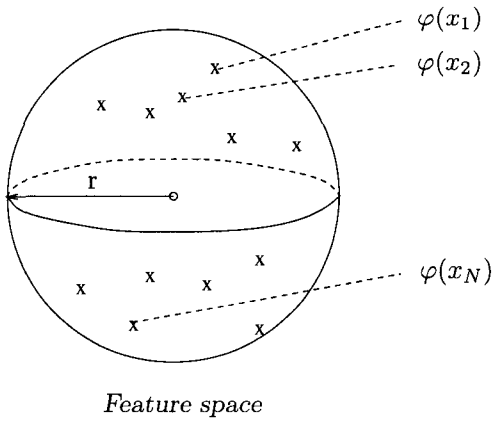


Fig. 2.11 *Smallest ball containing the points $\varphi(x_1), ..., \varphi(x_N)$ in the feature space. The radius r follows from a QP problem. Together with the margin this number r plays a role in the bound on the VC dimension for SVM classifiers.*

The radius r of the ball in the feature space can be computed as follows. Consider as optimization problem in the primal space

$$\left[\boxed{\text{P}}: \quad \min_{r,q} \quad r \qquad \text{such that} \quad \|\varphi(x_k) - q\|_2 < r, \ k = 1, 2, ..., N \right] \tag{2.37}$$

where q is a point inside the ball to be determined. The optimization problem expresses that all points should be contained within the ball with radius r in the feature space. The Lagrangian is

$$\mathcal{L}(r, q, \lambda) = r^2 - \sum_{k=1}^{N} \lambda_k (r^2 - \|\varphi(x_k) - q\|_2^2) \tag{2.38}$$

with positive Lagrange multipliers λ_k. After taking the conditions for op-

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

timality one obtains the following dual QP problem:

$$\left[\begin{array}{lll} \boxed{\mathrm{D}}: & \max\limits_{\lambda} & -\sum\limits_{k,l=1}^{N} K(x_k,x_l)\,\lambda_k\lambda_l + \sum\limits_{k=1}^{N}\lambda_k K(x_k,x_k) \\ & \text{such that} & \sum\limits_{k=1}^{N}\lambda_k = 1 \\ & & \lambda_k \geq 0\ , k=1,...,N \end{array}\right] \tag{2.39}$$

where the kernel trick $K(x_k,x_l) = \varphi(x_k)^T\varphi(x_l)$ has been applied again. Furthermore one also obtains $q = \sum_{k=1}^{N}\lambda_k\varphi(x_k)$.

A remark that should be made concerning the VC bound is that one has experienced that the bound (2.36) is often conservative. Nevertheless it can also be sufficiently indicative. Sharper bounds have been derived in terms of covering numbers and entropy numbers and can be expressed in terms of the eigenvalues of the Gram matrix as shown in [206; 207] and Rademacher complexity [15; 159]. A recent overview is given by Herbrich in [111] with results in the VC, PAC and luckiness framework, algorithmic stability and compression bounds and a PAC-Bayesian framework. For further reading on the problem of learning and generalization and VC theory, see also [11; 51; 285].

2.5 SVMs for function estimation

2.5.1 *SVM for linear function estimation*

In addition to classification, the support vector methodology has also been introduced for linear and nonlinear function estimation problems [279]. Let us discuss first the linear function estimation case. Consider regression in the set of linear functions

$$f(x) = w^T x + b \tag{2.40}$$

with N given training data with input values $x_k \in \mathbb{R}^n$ and output values $y_k \in \mathbb{R}$. For empirical risk minimization in the sense of Vapnik one employs the cost function

$$R_{\mathrm{emp}} = \frac{1}{N}\sum_{k=1}^{N}|y_k - w^T x_k - b|_\epsilon \tag{2.41}$$

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

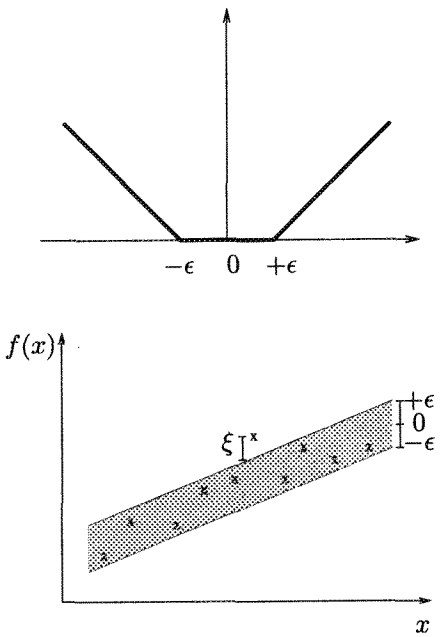


Fig. 2.12 *(Top) Vapnik ϵ-insensitive loss function for function estimation; (Bottom) Tube of ϵ-accuracy and points which cannot meet this accuracy, motivating the use of slack variables.*

with the so-called Vapnik's ϵ-insensitive loss function defined as

$$|y - f(x)|_\epsilon = \begin{cases} 0, & \text{if } |y - f(x)| \leq \epsilon \\ |y - f(x)| - \epsilon, & \text{otherwise} \end{cases} \tag{2.42}$$

which is shown in Fig. 2.12.

Estimation of a linear function is done then by formulating the following primal problem in w, b:

$$\left[\begin{array}{lll} \boxed{\text{P}}: & \min\limits_{w,b} J_P(w) = & \frac{1}{2} w^T w \\ & \text{such that} & y_k - w^T x_k - b \leq \epsilon, \quad k = 1, ..., N \\ & & w^T x_k + b - y_k \leq \epsilon, \quad k = 1, ..., N. \end{array} \right] \tag{2.43}$$

The value ϵ in the Vapnik ϵ-insensitive loss function is the accuracy that one requires for the approximation. The formulation (2.43) corresponds to a

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

case in which all training data points would belong to an ϵ-tube of accuracy. Of course when one chooses ϵ small, certain points will be outside this ϵ-tube and the problem (2.43) will become infeasible. Therefore additional slack variables ξ_k, ξ_k^* for $k = 1, ..., N$ are introduced. This is somewhat similar to the classifier case with overlapping distributions where also slack variables were introduced.

The problem (2.43) is modified into

$$\left[\begin{array}{lll} \boxed{\mathrm{P}}: \quad \min\limits_{w,b,\xi,\xi^*} J_{\mathrm{P}}(w,\xi,\xi^*) = & \frac{1}{2}w^T w + c\sum\limits_{k=1}^{N}(\xi_k + \xi_k^*) & \\ \text{such that} & y_k - w^T x_k - b \leq \epsilon + \xi_k \ , \ k = 1, ..., N & \\ & w^T x_k + b - y_k \leq \epsilon + \xi_k^* \ , \ k = 1, ..., N & \\ & \xi_k, \xi_k^* \geq 0 \ , \ k = 1, ..., N. & \end{array}\right] \tag{2.44}$$

The constant $c > 0$ determines the amount up to which deviations from the desired ϵ accuracy are tolerated.

The Lagrangian for this problem is

$$\begin{aligned} &\mathcal{L}(w, b, \xi, \xi^*; \alpha, \alpha^*, \eta, \eta^*) = \\ &\frac{1}{2}w^T w + c\sum_{k=1}^{N}(\xi_k + \xi_k^*) - \sum_{k=1}^{N}\alpha_k(\epsilon + \xi_k - y_k + w^T x_k + b) \\ &- \sum_{k=1}^{N}\alpha_k^*(\epsilon + \xi_k^* + y_k - w^T x_k - b) - \sum_{k=1}^{N}(\eta_k \xi_k + \eta_k^* \xi_k^*) \end{aligned} \tag{2.45}$$

with positive Lagrange multipliers $\alpha_k, \alpha_k^*, \eta_k, \eta_k^* \geq 0$. The saddle point of the Lagrangian is characterized by

$$\max_{\alpha,\alpha^*,\eta,\eta^*} \min_{w,b,\xi,\xi^*} \mathcal{L}(w, b, \xi, \xi^*; \alpha, \alpha^*, \eta, \eta^*) \tag{2.46}$$

with conditions for optimality:

$$\left\{\begin{array}{lll} \frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow & w = \sum\limits_{k=1}^{N}(\alpha_k - \alpha_k^*)x_k \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow & \sum\limits_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0 \\ \frac{\partial \mathcal{L}}{\partial \xi_k} = 0 & \rightarrow & c - \alpha_k - \eta_k = 0 \\ \frac{\partial \mathcal{L}}{\partial \xi_k^*} = 0 & \rightarrow & c - \alpha_k^* - \eta_k^* = 0. \end{array}\right. \tag{2.47}$$

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

The dual problem is again a QP problem:

$$\left[\begin{array}{lll} \boxed{\mathrm{D}}: \quad \max_{\alpha,\alpha^*} J_{\mathrm{D}}(\alpha,\alpha^*) = & -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k-\alpha_k^*)(\alpha_l-\alpha_l^*)\,x_k^T x_l & \\ & -\epsilon\sum_{k=1}^{N}(\alpha_k+\alpha_k^*)+\sum_{k=1}^{N} y_k(\alpha_k-\alpha_k^*) & \\ \text{such that} & \sum_{k=1}^{N}(\alpha_k-\alpha_k^*)=0 & \\ & \alpha_k,\alpha_k^*\in[0,c]. & \end{array}\right] \tag{2.48}$$

The SVM in primal weight space for linear function estimation is $f(x) = w^T x + b$. With $w = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)x_k$ this becomes in the dual space:

$$f(x) = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)x_k^T x + b \tag{2.49}$$

where α_k, α_k^* are the solution to the QP problem (2.48). The bias term b follows from the complementarity KKT conditions.

The properties of this solution are comparable with the results on classification. The solution is global and unique. Many of the elements in the solution vector will be equal to zero, which gives us again a sparseness property. In this linear case one can equally well solve the primal problem as the dual problem. The former is parametric while the latter is non-parametric. The size of the dual problem is independent of the dimension of the input space, but depends on the number of training data points.

2.5.2 *SVM for nonlinear function estimation*

Linear support vector regression can now be extended from the linear to the nonlinear case, again by application of the kernel trick.

In the primal weight space the model takes the form

$$f(x) = w^T\varphi(x) + b, \tag{2.50}$$

with given training data $\{x_k, y_k\}_{k=1}^N$ and $\varphi(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ a mapping to a high dimensional feature space which can be infinite dimensional and is only implicitly defined. Note that in this nonlinear case the vector w can

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

also become infinite dimensional. The optimization problem in the primal weight space becomes

$$\left[\begin{array}{ll} \boxed{\mathrm{P}}: \min_{w,b,\xi,\xi^*} J_{\mathrm{P}}(w,\xi,\xi^*) = \frac{1}{2}w^T w + c\sum_{k=1}^{N}(\xi_k + \xi_k^*) & \\ \text{such that} & y_k - w^T\varphi(x_k) - b \leq \epsilon + \xi_k \ , \ k = 1,...,N \\ & w^T\varphi(x_k) + b - y_k \leq \epsilon + \xi_k^* \ , \ k = 1,...,N \\ & \xi_k, \xi_k^* \geq 0 \ , \ k = 1,...,N. \end{array}\right] \tag{2.51}$$

In fact one can formally replace here x from the linear function estimation case by $\varphi(x)$.

After taking the Lagrangian and conditions for optimality one obtains the following dual problem:

$$\left[\begin{array}{ll} \boxed{\mathrm{D}}: \max_{\alpha,\alpha^*} J_{\mathrm{D}}(\alpha,\alpha^*) = & -\frac{1}{2}\sum_{k,l=1}^{N}(\alpha_k - \alpha_k^*)(\alpha_l - \alpha_l^*)\, K(x_k,x_l) \\ & -\epsilon\sum_{k=1}^{N}(\alpha_k + \alpha_k^*) + \sum_{k=1}^{N} y_k(\alpha_k - \alpha_k^*) \\ \text{such that} & \sum_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0 \\ & \alpha_k, \alpha_k^* \in [0,c]. \end{array}\right] \tag{2.52}$$

Here the kernel trick has been applied with $K(x_k,x_l) = \varphi(x_k)^T\varphi(x_l)$ for $k,l = 1,...,N$. The dual representation of the model becomes

$$f(x) = \sum_{k=1}^{N}(\alpha_k - \alpha_k^*)K(x,x_k) + b \tag{2.53}$$

where α_k, α_k^* are the solution to the QP problem (2.74) and b follows from the complementarity KKT conditions.

The solution to the QP problem is global and unique provided that the chosen kernel function is positive definite. As in the classifier case, the solution is sparse. The size of the QP problem does not depend on the dimension of the input space. While in the linear case one might equally well solve the primal problem, this is no longer the case for the nonlinear function

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

estimation problem as the w vector might become infinite dimensional and often $\varphi(\cdot)$ is not explicitly known. In comparison with the nonlinear SVM classifier case, the nonlinear SVM regression formulation contains more tuning parameters. In the case of an RBF kernel σ, c, ϵ are to be considered as additional tuning parameters which do not follow as the solution to the QP problem but should be determined in another way, e.g. based on cross-validation or by applying VC bounds for the regression case. In Fig. 2.13 and Fig. 2.14 some illustrations are given on the effect of changing ϵ for a noisy sinc function and the corresponding support vectors using the SVM Matlab toolbox by Steve Gunn.

In order to have a better control upon the number of support vectors one has proposed other formulations such as the *ν-tube support vector regression* [205]. In this method the primal objective function is modified as follows:

$$\frac{1}{2}w^T w + c\left(\nu\epsilon + \frac{1}{N}\sum_{k=1}^{N}(\xi_k + \xi_k^*)\right). \tag{2.54}$$

In this approach one can control by ν the fraction of support vectors that is allowed to lie outside the tube, which is directly proportional to the number of support vectors in an asymptotic sense.

2.5.3 *VC bound on generalization error*

For the nonlinear function estimation case VC bounds have also been derived [41; 42; 206; 281]. For the empirical risk with squared loss function

$$R_{\text{emp}}(\theta) = \frac{1}{N}\sum_{k=1}^{N}(y_k - f(x_k;\theta))^2 \tag{2.55}$$

and the predicted risk

$$R(\theta) = \int (y - f(x;\theta))^2 p(x,y)dxdy \tag{2.56}$$

one has the following VC bound

$$R(\theta) \leq R_{\text{emp}}(\theta)\left(1 - c\sqrt{\frac{h(\ln(aN/h)+1) - \ln\eta}{N}}\right)_+^{-1} \tag{2.57}$$

where h denotes the VC dimension of the set of approximating functions. In [42] $a = c = 1$ has been chosen for the constants. This bound holds

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com
by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

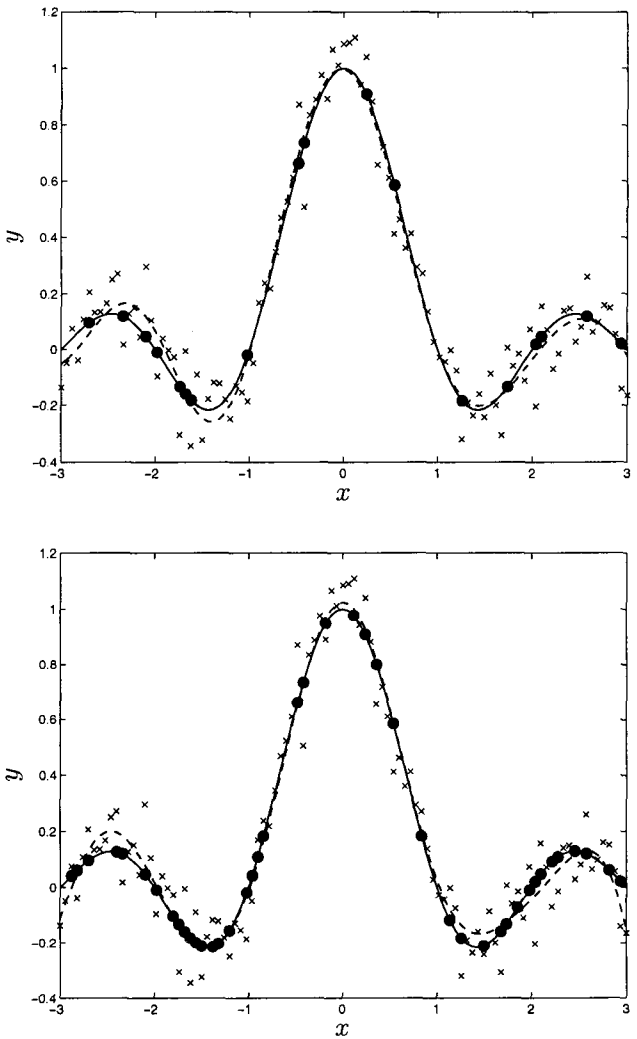


Fig. 2.13 *Illustration of SVM with Vapnik ϵ-insensitive loss function on a noisy sinc function (zero mean Gaussian noise with standard dev. equal to 0.01) with (Top) $c = 100$ and $\epsilon = 0.15$ and RBF kernel with $\sigma = 0.8$ giving 18 support vectors and (Bottom) $\epsilon = 0.15/2$ giving 43 support vectors. True sinc function (solid line); SVM output (dashed line).*

Least Squares Support Vector Machines Downloaded from www.worldscientific.com
by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

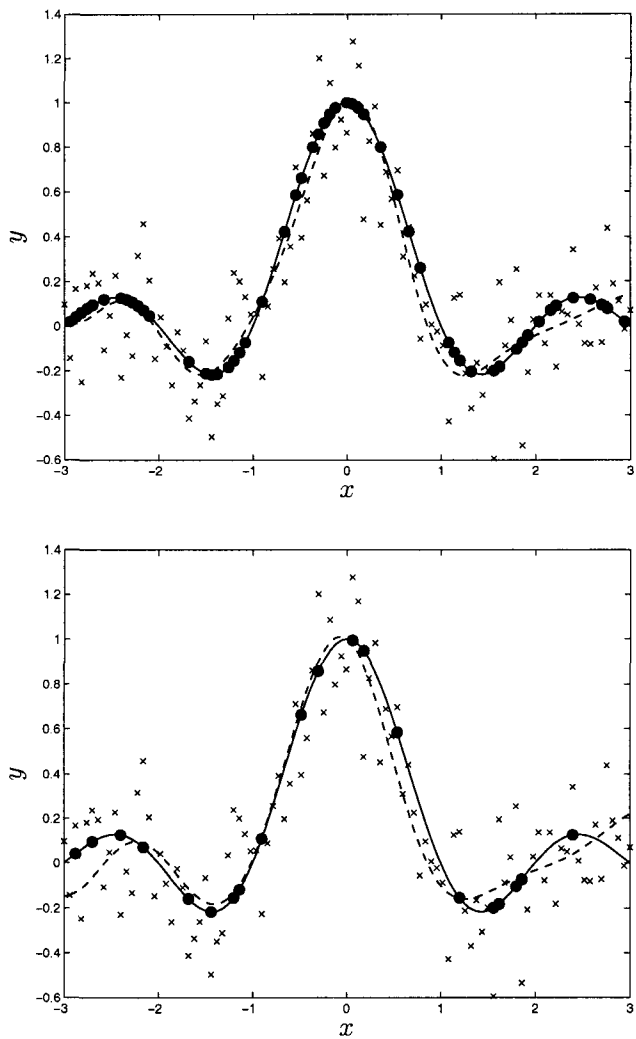


Fig. 2.14 *Illustration of SVM with Vapnik ϵ-insensitive loss function on a noisy sinc function (zero mean Gaussian noise with standard dev. equal to 0.05) with (Top) $c = 100$ and $\epsilon = 0.15$ and RBF kernel with $\sigma = 0.8$ resulting into 54 support vectors and (Bottom) $\epsilon = 0.3$ giving 20 support vectors. True sinc function (solid line); SVM output (dashed line).*

with probability $1 - \eta$. The notation $(x)_+$ means $(x)_+ = x$ if $x > 0$ and 0 otherwise.

The estimated risk can be expressed as

$$R_{\text{est}} = g(h, N)\frac{1}{N}\sum_{k=1}^{N}(y_k - f(x_k;\theta))^2 \tag{2.58}$$

where $g(h, N)$ is a correcting function.

After bringing the formula in this form it is straightforward to establish the link with well-known classical criteria in statistics (see Vapnik [281]) such as

- *Finite prediction error* (FPE) (Akaike) [5]

$$g(d, N) = \frac{1 + d/N}{1 - d/N} \tag{2.59}$$

- *Generalized cross-validation* (GCV) (Craven & Wahba) [49]

$$g(d, N) = \frac{1}{\left(1 - \frac{d}{N}\right)^2} \tag{2.60}$$

- *Shibata's model selector* (SMS) [214]

$$g(d, N) = 1 + 2\frac{d}{N} \tag{2.61}$$

- *Schwarz criteria* (MDL criteria) [209]

$$g(d, N) = 1 + \frac{\frac{d}{N}\log N}{2\left(1 - \frac{d}{N}\right)} \tag{2.62}$$

with d the number of free parameters for a model which is linear in the parameters.

Recent work on the topic of generalization and the mathematical foundations of learning theory have been given by Cucker & Smale in [54]. In this work bounds on the generalization error are derived with a careful analysis of the approximation error and sample error, with results specified to e.g. Sobolev spaces and reproducing kernel Hilbert spaces.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

2.6 Modifications and extensions

2.6.1 *Kernels*

Kernels from kernels

In the previous Sections some typical choices of positive definite kernels were discussed such as the linear, polynomial, RBF and MLP kernel. In classification problems one frequently employs the linear, polynomial and RBF kernel. In nonlinear function estimation and nonlinear modelling problems one often uses the RBF kernel.

However, there are many other opportunities to go beyond such basic choices. A first extension is that it is allowed to do many operations on positive definite kernels such that they still remain positive definite. For symmetric positive definite kernels the following operations are allowed as explained for example by Cristianini & Shawe-Taylor in [51]:

- $K(x,z) = aK_1(x,z) \quad (a > 0)$
- $K(x,z) = K_1(x,z) + b \quad (b > 0)$
- $K(x,z) = x^T P z \quad (P = P^T > 0)$
- $K(x,z) = K_1(x,z) + K_2(x,z)$
- $K(x,z) = K_1(x,z)K_2(x,z)$
- $K(x,z) = f(x)f(z)$
- $K(x,z) = K_3(\phi(x), \phi(z))$
- $K(x,z) = p_+(K_4(x,z))$
- $K(x,z) = \exp(K_4(x,z))$

where $a, b \in \mathbb{R}^+$ and K_1, K_2, K_3, K_4 are symmetric positive definite kernel functions, $f(\cdot) : \mathbb{R}^n \to \mathbb{R}$, $\phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_h}$ and $p_+(\cdot)$ is a polynomial with positive coefficients.

One may also normalize kernels. For the linear kernel one can take:

$$K(x,z) = \frac{x^T z}{\|x\|_2 \|z\|_2} = \cos(\theta_{\{x,z\}}) \tag{2.63}$$

where $\theta_{\{x,z\}}$ denotes the angle between the vectors x, z in the input space,

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

or in general

$$\begin{aligned} \tilde{K}(x,z) &= \frac{K(x,z)}{\sqrt{K(x,x)}\sqrt{K(z,z)}} \\ &= \cos(\theta_{\{\varphi(x),\varphi(z)\}}) \end{aligned} \tag{2.64}$$

where $\tilde{K}$ is the normalized kernel [108] and $\theta_{\{\varphi(x),\varphi(z)\}}$ is the angle between $\varphi(x), \varphi(z)$ in the feature space. Also note that one can express $d(\varphi(x), \varphi(z))$ as the Euclidean distance between $\varphi(x), \varphi(z)$ in the feature space as

$$d(\varphi(x), \varphi(z))^2 = \|\varphi(x) - \varphi(z)\|_2^2 = K(x,x) + K(z,z) - 2K(x,z). \tag{2.65}$$

Hence instead of working with $d(x,z)$ which is the Euclidean distance between x, z one may take the RBF kernel $K(x,z) = \exp(-\|x-z\|_2^2/\sigma^2) = \exp(-d(x,z)^2/\sigma^2)$ such that:

$$\begin{aligned} \tilde{K}(x,z) &= \exp\left(-\frac{K(x,x) + K(z,z) - 2K(x,z)}{\sigma^2}\right) \\ &= \exp\left(-\frac{d(\varphi(x),\varphi(z))^2}{\sigma^2}\right). \end{aligned} \tag{2.66}$$

The normalization of the kernel (2.64) may be considered as a special case of a conformal transformation of the kernel

$$\tilde{K}(x,z) = c(x)K(x,z)c(z) \tag{2.67}$$

with factor $c(x)$. This was successfully applied by Amari [7] for modifying the kernel in a data dependent way in order to increase the margin by enlarging the spatial resolution around the decision boundary. Also differential geometrical interpretations have been given at this point by showing that the metric can be directly derived from the kernel. The link between the choice of the kernel and differential geometry has also been investigated by Schölkopf *et al.* [203] and Burges in [202]. From this study one finds e.g. for which values of κ_1, κ_2 the MLP kernel $\tanh(\kappa_1 x^T z + \kappa_2)$ is positive definite.

Compactly supported kernels

Towards computational methods it can be an advantage to work with a compactly supported kernel, e.g. in the case of an RBF kernel one may try to cut-off the kernel in order to create zeros (sparseness) in the kernel matrix.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

However, this will often result in a kernel which is no longer positive definite and should be avoided. Therefore, in [93] a general way has been discussed to obtain a compactly supported kernel from a given kernel without loosing positive definiteness and has been applied in [104].

Training a classical neural network by SVM methodology

While SVM methodology largely makes use of the fact that one chooses a kernel function without worrying about the underling feature space map $\varphi(\cdot)$, unless showing that it exists by taking a positive definite kernel, one may also explicitly choose $\varphi(\cdot)$ and compute $K(x,z) := \varphi(x)^T\varphi(z)$. In [234] this idea was applied to show that one can also train classical MLP neural networks by SVM methodology. A classical MLP classifier of the form

$$y(x) = \text{sign}[w^T \tanh(Vx + \beta)] \tag{2.68}$$

was interpreted as an SVM classifier in the primal weight space by defining $\varphi(x) = \tanh(Vx + \beta)$ where the feature space explicitly corresponds to the hidden layer space of the MLP classifier. The kernel then becomes $K(x,z) = \tanh(Vx + \beta)^T \tanh(Vz + \beta)$. In this case all the elements of the hidden layer matrix V and the bias vector β are to be considered as tuning parameters, which shows the computational advantage and powerful representation of choosing e.g. an RBF kernel in SVM which has only one single tuning parameter.

String kernels and textmining

Instead of letting kernels operate on real numbers one can also go far beyond such limitations. Special type of kernels have been considered for example that can operate on strings as shown by Cristianini [51] and Haussler [108]. Furthermore, in applications of text categorization for classification of documents the following choice of a kernel is meaningful [51]

$$\varphi_i(x) = a\,\text{TF}_i \log(\text{IDF}_i) \tag{2.69}$$

and in accordance with the information retrieval literature. Here TF_i denotes the number of occurrences of a term i in the document x, IDF_i the number of documents containing the term and a a normalization constant which is chosen such that $\|\varphi(x)\|_2 = 1$. The vector $\varphi(x)$ is typically very

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

large and sparse. The kernel is chosen then as $K(x,z) := \varphi(x)^T\varphi(z)$ and again positive definite by construction.

Bioinformatics applications

Special type of kernels have also been developed towards certain bioinformatics applications [206; 303]. For example in order to recognize translation initiation sites from which coding starts and to determine which parts of a sequence will be translated one may employ the kernel

$$K(x,z) = \left(\sum_{p=1}^{l} \mathrm{win}_p(x,z)\right)^{d_2}$$
$$\text{with} \qquad \mathrm{win}_p(x,z) = \left(\sum_{j=-l}^{+l} v_j \mathrm{match}_{p+j}(x,z)\right)^{d_1} \tag{2.70}$$

where $\mathrm{match}_{p+j}(x,z)$ is defined to be 1 for matching nucleotides at position $p+j$ and zero otherwise, for the DNA four letter alphabet of nucleotides $\{A,C,G,T\}$. This kernel is so-called *locality improved* as it emphasizes local correlations. The window scores computed with win_p are summed over the whole length of the sequence. Correlations between windows of order up to d_2 are taking into account. At each sequence position one compares two sequences locally, within a small window length $2l+1$ around that position. Matching nucleotides are summed and multiplied with weights v_j having increasing values towards the center of the window.

For microarray data classification the use of traditional linear, polynomial and RBF kernels has been investigated [31; 89; 297].

Optimal kernels

In general the problem of which is the most suitable kernel for a particular application or problem is still an open problem up till now. Attempts in this direction are [121; 52] using Fisher kernels and optimizing kernel alignment and kernel-target alignment.

2.6.2 *Extension to other convex cost functions*

In the previous Sections commonly used cost functions were discussed as originally proposed by Vapnik. It is possible to generalize the results of

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

SVM regression to any convex cost function.

According to Vapnik [282] and Smola & Schölkopf [219], consider a loss function

$$L_\epsilon(y - f(x)) = \begin{cases} 0, & \text{if } |y - f(x)| \leq \epsilon \\ L(y - f(x)) - \epsilon, & \text{otherwise} \end{cases} \tag{2.71}$$

where $L(\cdot)$ is convex. One can formulate then the primal problem

$$\left[\begin{array}{lll} \boxed{\text{P}}: & \min\limits_{w,b,\xi,\xi^*} & \frac{1}{2}w^T w + c\sum\limits_{k=1}^{N}(L(\xi_k) + L(\xi_k^*)) \\ & \text{such that} & y_k - w^T\varphi(x_k) - b \leq \epsilon + \xi_k \\ & & w^T\varphi(x_k) + b - y_k \leq \epsilon + \xi_k^* \\ & & \xi_k, \xi_k^* \geq 0 \end{array}\right] \tag{2.72}$$

where ξ, ξ^* are slack variables. The Lagrangian for this problem is

$$\begin{aligned} &\mathcal{L}(w, b, \xi, \xi^*; \alpha, \alpha^*, \eta, \eta^*) = \\ &\frac{1}{2}w^T w + c\sum_{k=1}^{N}(L(\xi_k) + L(\xi_k^*)) - \sum_{k=1}^{N}\alpha_k(\epsilon + \xi_k - y_k + w^T\varphi(x_k) + b) \\ &\quad - \sum_{k=1}^{N}\alpha_k^*(\epsilon + \xi_k^* + y_k - w^T\varphi(x_k) - b) - \sum_{k=1}^{N}(\eta_k\xi_k + \eta_k^*\xi_k^*) \end{aligned} \tag{2.73}$$

with Lagrange multipliers $\alpha_k, \alpha_k^*, \eta_k, \eta_k^* \geq 0$ for $k = 1, ..., N$.

One obtains the following dual problem

$$\left[\begin{array}{lll} \boxed{\text{D}}: & \max\limits_{\alpha,\alpha^*,\eta,\eta^*} & J_{\text{D}}(\alpha, \alpha^*, \eta, \eta^*) \\ & \text{such that} & \sum\limits_{k=1}^{N}(\alpha_k - \alpha_k^*) = 0 \\ & & cL'(\xi_k) - \alpha_k - \eta_k = 0\ ,\ k = 1, ..., N \\ & & cL'(\xi_k^*) - \alpha_k^* - \eta_k^* = 0\ ,\ k = 1, ..., N \\ & & \alpha_k, \alpha_k^*, \eta_k, \eta_k^* \geq 0\ ,\ k = 1, ..., N \end{array}\right] \tag{2.74}$$

where $J_{\text{D}}(\alpha, \alpha^*, \eta, \eta^*)$ is the objective function for this dual problem obtained after elimination of the primal variables. In these expressions one further needs to eliminate $L'(\xi_k)$. This is possible for example for $L(\xi) = \xi$ (Vapnik ϵ-insensitive loss function), $L(\xi) = \xi^2$ (quadratic loss function with

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

ϵ-zone) and Huber loss function with ϵ-zone as explained in [282]. The Huber loss function is well-known in the area of robust statistics. For these problems the dual variables η, η^* can be eliminated from the problem. In [219] the results have been further generalized to any convex cost function.

From these equations one can see that the obtained *sparseness* for the model is due to the ϵ-zone around the origin of the loss function. In this region one has $L'(\cdot) = 0$ such that one obtains $\alpha_k = -\eta$, $\alpha_k^* = -\eta^*$ and because $\alpha_k, \alpha_k^*, \eta_k, \eta_k^* \geq 0$, one obtains then $\alpha_k = 0$, $\alpha_k^* = 0$.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

2.6.3 *Algorithms*

Interior point algorithms

The dual problems in SVM formulations can usually be cast in the form

$$\begin{array}{ll} \min & \frac{1}{2}q(\alpha) + c^T\alpha \\ \text{such that} & A\alpha = b \\ & l \leq \alpha \leq u \end{array} \tag{2.75}$$

where α denotes here the vector of all dual variables, $c, \alpha, l, u \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ and $q(\alpha)$ a convex function in α.

An important class of methods for solving convex programs are *interior point algorithms* [86; 173; 174; 190; 261]. Since the seminal work of Karmarkar in linear programming, this approach has been successfully extended to nonlinear programs. Several techniques have been developed for interior point algorithms including e.g. central path methods, path following methods and potential reduction methods.

In [219] the implementation of a path-following method has been investigated for SVMs by Smola. In (2.75) one first translates the inequality constraints into equality constraints by adding slack variables g, t as follows

$$\begin{array}{ll} \min & \frac{1}{2}q(\alpha) + c^T\alpha \\ \text{such that} & A\alpha = b \\ & \alpha - g = l \\ & \alpha + t = u \\ & g, t \geq 0. \end{array} \tag{2.76}$$

The Wolfe dual to (2.76) is

$$\begin{array}{ll} \max & \frac{1}{2}(q(\alpha) - \nabla q(\alpha)^T \alpha) + b^T y + l^T z - u^T s \\ \text{such that} & \frac{1}{2}\nabla q(\alpha) + c - (Ay)^T + s = z \\ & s, z \geq 0 \end{array} \tag{2.77}$$

together with

$$\begin{array}{rcl} g_i z_i & = & 0 \\ s_i t_i & = & 0 \ , \ i = 1, ..., n \end{array} \tag{2.78}$$

from the KKT conditions. For the interior point algorithm the latter is modified then into

$$\begin{array}{rcl} g_i z_i & = & \mu \\ s_i t_i & = & \mu \ , \ i = 1, ..., n \end{array} \tag{2.79}$$

and the problem is iteratively solved by gradually decreasing the value of μ. The above system of equations is linearized and the resulting equations are solved by a predictor-corrector approach until the duality gap is small enough. This linearization gives the following after neglecting terms in Δ^2:

$$\begin{array}{lll} A\Delta\alpha & = b - A\alpha & =: \rho \\ \Delta\alpha - \Delta g & = l - \alpha + g & =: \nu \\ \Delta\alpha + \Delta t & = u - \alpha - t & =: \tau \\ (A\Delta y)^T + \Delta z - \Delta s - \frac{1}{2}\nabla^2 q(\alpha)\Delta\alpha & = c - (Ay)^T + \frac{1}{2}\nabla q(\alpha) + s - z & =: \sigma \\ g_{\text{inv}_i} z_i \Delta g_i + \Delta z_i & = \mu g_{\text{inv}_i} - z_i - g_{\text{inv}_i}\Delta g_i \Delta z_i & =: \gamma_{z_i} \\ t_{\text{inv}_i} s_i \Delta t_i + \Delta s_i & = \mu t_{\text{inv}_i} - s_i - t_{\text{inv}_i}\Delta t_i \Delta s_i & =: \gamma_{s_i} \end{array} \tag{2.80}$$

where $g_{\text{inv}} = [1/g_1; ...; 1/g_n]$ and similarly for t_{inv}, z_{inv}, s_{inv}. Solving for $\Delta g, \Delta t, \Delta z, \Delta s$ one gets

$$\begin{array}{rcl} \Delta g_i & = & z_{\text{inv}_i} g_i (\gamma_{z_i} - \Delta z_i) \\ \Delta t_i & = & s_{\text{inv}_i} t_i (\gamma_{s_i} - \Delta s_i) \\ \Delta z_i & = & g_{\text{inv}_i} z_i (\hat{\nu}_i - \Delta\alpha_i) \\ \Delta s_i & = & t_{\text{inv}_i} s_i (\Delta\alpha_i - \hat{\tau}_i) \end{array} \tag{2.81}$$

where $\hat{\nu}_i = \nu_i - z_{\text{inv}_i} g_i \gamma_{z_i}$, $\hat{\tau}_i = \tau_i - s_{\text{inv}_i} t_i \gamma_{s_i}$ for $i = 1, ..., n$. One also obtains the *reduced KKT system*

$$\begin{bmatrix} -(\frac{1}{2}\nabla^2 q(\alpha) + D) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta\alpha \\ \Delta y \end{bmatrix} = \begin{bmatrix} \sigma - r \\ \rho \end{bmatrix} \tag{2.82}$$

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

where $D = \text{diag}[g_{\text{inv}_1} z_1 + t_{\text{inv}_1} s_1; ...; g_{\text{inv}_n} z_n + t_{\text{inv}_n} s_n]$ and $r_i = g_{\text{inv}_i} z_i \hat{\nu}_i - t_{\text{inv}_i} s_i \hat{\tau}_i$.

In the predictor step one solves the system (2.81) and (2.82) with $\nu = 0$ and all Δ terms on the right hand side set to zero, i.e. $\gamma_z = z$, $\gamma_s = s$. The values in Δ are substituted back into the definitions for γ_z and γ_s and (2.81) and (2.82) are solved again in the corrector step. The matrix needs to be inverted then only once in a given iteration step. To ensure that the variables meet the positivity constraints, a careful choice of the steplength ξ is needed. The value of μ is often chosen as

$$\mu = \frac{g^T z + s^T t}{2n} \left(\frac{\xi - 1}{\xi + 10} \right)^2 . \tag{2.83}$$

The reduced KKT system is often additionally regularized in order to further improve the behaviour of the algorithm [262]. Also a good initialization of the algorithm will improve its performance. For further details on this interior point algorithm the reader may consult [219; 262].

Large data sets: chunking, decomposition and others

The interior point method as described here is only suitable for typically about 2000 data points (depending on the computer memory) assuming that the matrices are stored. A possible way for applying SVMs to larger data sets is to start with a first *chunk* of the data set that fits into memory. One trains an SVM on this chunk, keeps the support vectors and deletes the non-support vector data. This removed part can be replaced then by a new part of the large training set, where one selects data on which the current SVM makes an error. The system is retrained and one iterates until the KKT conditions are satisfied for all training data.

In *decomposition* methods [142] one works with a fixed size subset while freezing the other variables, which is often more convenient. Results on chunking and decomposition were made by Vapnik [279], Osuna *et al.* [175], Joachims [202], Lin [142] and others. An extreme form of decomposition is sequential minimal optimization (SMO) [182] where very small subsets of size 2 are selected. In this case one can find an analytic expression for the solution to the two-datapoint sub-problem.

On the other hand other optimization algorithms have also been used such as SOR (Successive OverRelaxation) on massive data sets with millions of data points by Mangasarian & Musicant in [155]. Within the class of

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

iterative methods conjugate gradient methods perform well, see e.g. Kaufman in [202]. These methods are suitable for processing large data sets as well.

2.6.4 *Parametric versus non-parametric approaches*

The original Vapnik formulation for SVMs emphasizes primal-dual interpretations and makes use of the kernel trick. In recent years one has also recognized that this kernel trick can be applied in a more general fashion. Often existing linear techniques can be kernelized into nonlinear version. However, sometimes there might arise confusion with respect to parametric versus non-parametric issues at this point.

As explained in the previous Chapter the link between parametric models of radial basis function (RBF) networks and their parametric version could be understood by taking the centers of the Gaussians at the data points itself. However, in this case one should keep in mind that the unknown parameter vector size is fixed and the centers accidently coincide with the training data points. One could take a parametric kernel model as follows:

$$f(x;w) = \sum_{i=1}^{N} w_i K(x, x_i) \tag{2.84}$$

with $w \in \mathbb{R}^N$ a fixed size parameter vector and estimate the parameter vector as if it were a non-kernel based model (which traditionally belongs to the class of non-parametric methods). The notation w is chosen here instead of α in order to avoid confusion with SVM models in the dual space of Lagrange multipliers. Also note that there is no need at all at this point that the kernel function should be positive definite and satisfy the Mercer condition, because in (2.84) one may choose in principle *any* parameterization that one would like.

As explained in the previous chapter one could proceed then as with the training of classical MLPs where one considers a weight decay term and find w from

$$\min_w J(w) = \|w\|_2^2 + \sum_{i=1}^{N} \left(y_i - \sum_{i=1}^{N} w_i K(x, x_i) \right)^2 . \tag{2.85}$$

The technique of automatic relevance determination (ARD) has then been

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

used within a Bayesian inference context in order to assess the importance of the weights and hence on the data points, because by construction of the model each weight w_i corresponds to a data point $x_i \in \mathbb{R}^n$ (which is not the case e.g. in MLP neural networks). This method is related to what has been called the *relevance vector machine* in [253]. From a neural networks perspective such a model should rather be considered as a radial basis function network than as a support vector machine.

One can achieve sparseness for this model also by taking another regularization term

$$\min_w J(w) = \|w\|_1 + \sum_{i=1}^{N} \left(y_i - \sum_{i=1}^{N} w_i K(x, x_i) \right)^2 \tag{2.86}$$

where the 1-norm $\|w\|_1 = \sum_{i=1}^{N} |w_i|$ is used. This regularization term corresponds to a Laplacian prior when interpreted within a Bayesian inference context. Such methods are related to methods of sparse approximation [38; 95] and the Lasso and shrinkage estimators [107] in statistics.

Presently many methods are called kernel machines. However, not all of these possess the same primal-dual interpretations of SVMs as proposed by Vapnik or make use of the kernel trick. LP kernel machine [206; 225] formulations are based on the Vapnik ϵ-insensitive loss function. One optimizes

$$R_{\text{reg}}(w) = \frac{1}{N} \sum_{i=1}^{N} |w_i| + c \frac{1}{N} \sum_{i=1}^{N} |y_i - f(x_i)|_\epsilon \tag{2.87}$$

which leads to the following LP problem

$$\begin{aligned} \min \quad & \frac{1}{N} \sum_{i=1}^{N} (w_i + w_i^*) + \frac{c}{N} \sum_{i=1}^{N} (\xi_i + \xi_i^*) \\ \text{such that} \quad & \sum_{i=1}^{N} (w_i - w_i^*) K(x_i, x_j) + b - y_i \leq \epsilon + \xi_i \\ & y_i - \sum_{i=1}^{N} (w_i - w_i^*) K(x_i, x_j) - b \leq \epsilon + \xi_i^* \\ & w_i, w_i^*, \xi_i, \xi_i^* \geq 0 \end{aligned} \tag{2.88}$$

for $i, j = 1, ..., N$. Here w_i has been replaced by the positive variables w_i, w_i^* in order to avoid $|w_i|$ in the cost function. According to Smola in [225] there

Least Squares Support Vector Machines Downloaded from www.worldscientific.com by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.

is no computational advantage for this LP problem to compute the Wolfe dual. Methods for large scale LP problems are available in the optimization literature including simplex methods and interior point algorithms.

Least Squares Support Vector Machines Downloaded from www.worldscientific.com
by WEIZMANN INSTITUTE OF SCIENCE on 04/06/16. For personal use only.