6

Support Vector Machines

The material covered in the first five chapters has given us the foundation on which to introduce Support Vector Machines, the learning approach originally developed by Vapnik and co-workers. Support Vector Machines are a system for efficiently training the linear learning machines introduced in Chapter 2 in the kernel-induced feature spaces described in Chapter 3, while respecting the insights provided by the generalisation theory of Chapter 4, and exploiting the optimisation theory of Chapter 5. An important feature of these systems is that, while enforcing the learning biases suggested by the generalisation theory, they also produce 'sparse' dual representations of the hypothesis, resulting in extremely efficient algorithms. This is due to the Karush–Kuhn–Tucker conditions, which hold for the solution and play a crucial role in the practical implementation and analysis of these machines. Another important feature of the Support Vector approach is that due to Mercer's conditions on the kernels the corresponding optimisation problems are convex and hence have no local minima. This fact, and the reduced number of non-zero parameters, mark a clear distinction between these system and other pattern recognition algorithms, such as neural networks. This chapter will also describe the optimisation required to implement the Bayesian learning strategy using Gaussian processes.

6.1 Support Vector Classification

The aim of Support Vector classification is to devise a computationally efficient way of learning 'good' separating hyperplanes in a high dimensional feature space, where by 'good' hyperplanes we will understand ones optimising the generalisation bounds described in Chapter 4, and by 'computationally efficient' we will mean algorithms able to deal with sample sizes of the order of 100 000 instances. The generalisation theory gives clear guidance about how to control capacity and hence prevent overfitting by controlling the hyperplane margin measures, while optimisation theory provides the mathematical techniques necessary to find hyperplanes optimising these measures. Different generalisation bounds exist, motivating different algorithms: one can for example optimise the maximal margin, the margin distribution, the number of support vectors, etc. This chapter will consider the most common and well-established approaches which reduce the problem to minimising the norm of the weight vector. At the end of

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

the chapter we will provide pointers to other related algorithms, though since research in this field is still in progress, we make no attempt to be exhaustive.

6.1.1 The Maximal Margin Classifier

The simplest model of Support Vector Machine, which was also the first to be introduced, is the so-called maximal margin classifier. It works only for data which are linearly separable in the feature space, and hence cannot be used in many real-world situations. Nonetheless it is the easiest algorithm to understand, and it forms the main building block for the more complex Support Vector Machines. It exhibits the key features that characterise this kind of learning machine, and its description is therefore crucial for understanding the more advanced systems introduced later.

Theorem 4.18 in Chapter 4 bounds the generalisation error of linear machines in terms of the margin $m_S(f)$ of the hypothesis f with respect to the training set S. The dimensionality of the space in which the data are separated does not appear in this theorem. The maximal margin classifier optimises this bound by separating the data with the maximal margin hyperplane, and given that the bound does not depend on the dimensionality of the space, this separation can be sought in any kernel-induced feature space. The maximal margin classifier forms the strategy of the first Support Vector Machine, namely to find the maximal margin hyperplane in an appropriately chosen kernel-induced feature space.

This strategy is implemented by reducing it to a convex optimisation problem: minimising a quadratic function under linear inequality constraints. First we note that in the definition of linear classifiers there is an inherent degree of freedom, due to the fact that the function associated with the hyperplane $(\mathbf{w}, b)$ does not change if we rescale the hyperplane to $(\lambda\mathbf{w}, \lambda b)$, for $\lambda \in \mathbb{R}^+$. There will, however, be a change in the margin as measured by the function output as opposed to the geometric margin. We refer to the margin of the function output as the *functional margin.* Theorem 4.18 involves the *geometric margin*, that is the functional margin of a normalised weight vector. Hence, we can equally well optimise the geometric margin by fixing the functional margin to be equal to 1 (hyperplanes with functional margin 1 are sometimes known as *canonical hyperplanes*) and minimising the norm of the weight vector. If $\mathbf{w}$ is the weight vector realising a functional margin of 1 on the positive point $\mathbf{x}^+$ and the negative point $\mathbf{x}^-$, we can compute its geometric margin as follows. Recall that a functional margin of 1 implies

$$\begin{aligned} \langle \mathbf{w} \cdot \mathbf{x}^+ \rangle + b &= +1, \\ \langle \mathbf{w} \cdot \mathbf{x}^- \rangle + b &= -1, \end{aligned}$$

while to compute the geometric margin we must normalise $\mathbf{w}$. The geometric

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

margin γ is then the functional margin of the resulting classifier

$$\begin{aligned}\gamma &= \frac{1}{2}\left(\left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|_2}\cdot \mathbf{x}^+\right\rangle - \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|_2}\cdot \mathbf{x}^-\right\rangle\right) \\ &= \frac{1}{2\|\mathbf{w}\|_2}\left(\langle \mathbf{w}\cdot\mathbf{x}^+\rangle - \langle \mathbf{w}\cdot\mathbf{x}^-\rangle\right) \\ &= \frac{1}{\|\mathbf{w}\|_2}.\end{aligned}$$

Hence, the resulting geometric margin will be equal to $1/\|\mathbf{w}\|_2$ and we have demonstrated the following result.

Proposition 6.1 *Given a linearly separable training sample*

$$S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell))$$

the hyperplane $(\mathbf{w},b)$ *that solves the optimisation problem*

$$\begin{array}{ll} \text{minimise}_{\mathbf{w},b} & \langle \mathbf{w}\cdot\mathbf{w}\rangle, \\ \text{subject to} & y_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b\right) \geq 1, \\ & i = 1, \dots, \ell, \end{array}$$

realises the maximal margin hyperplane with geometric margin $\gamma = 1/\|\mathbf{w}\|_2$.

We now consider how to transform this optimisation problem into its corresponding dual problem following the strategy outlined in Section 5.3. The primal Lagrangian is

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\langle \mathbf{w}\cdot\mathbf{w}\rangle - \sum_{i=1}^{\ell}\alpha_i\left[y_i\left(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b\right) - 1\right]$$

where $\alpha_i \geq 0$ are the Lagrange multipliers, as described in Chapter 5.

The corresponding dual is found by differentiating with respect to $\mathbf{w}$ and b, imposing stationarity,

$$\begin{aligned}\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{\ell} y_i\alpha_i\mathbf{x}_i = \mathbf{0}, \\ \frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} &= \sum_{i=1}^{\ell} y_i\alpha_i = 0,\end{aligned}$$

and resubstituting the relations obtained,

$$\begin{aligned}\mathbf{w} &= \sum_{i=1}^{\ell} y_i\alpha_i\mathbf{x}_i, \\ 0 &= \sum_{i=1}^{\ell} y_i\alpha_i,\end{aligned}$$

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

into the primal to obtain

$$\begin{aligned} L(\mathbf{w}, b, \boldsymbol{\alpha}) &= \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle - \sum_{i=1}^{\ell} \alpha_i \left[y_i \left(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \right) - 1 \right] \\ &= \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle - \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle + \sum_{i=1}^{\ell} \alpha_i \\ &= \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle . \end{aligned}$$

Remark 6.2 The first of the substitution relations shows that the hypothesis can be described as a linear combination of the training points: the application of optimisation theory naturally leads to the dual representation introduced in Chapter 2. The dual representation is also required for the use of kernels.

We have therefore shown the main part of the following proposition, which follows from Proposition 6.1.

Proposition 6.3 *Consider a linearly separable training sample*

$$S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell)),$$

and suppose the parameters $\boldsymbol{\alpha}^*$ *solve the following quadratic optimisation problem:*

$$\begin{array}{ll} \textit{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle, \\ \textit{subject to} & \sum_{i=1}^{\ell} y_i \alpha_i = 0, \\ & \alpha_i \geq 0,\ i = 1, \dots, \ell. \end{array} \qquad (6.1)$$

Then the weight vector $\mathbf{w}^* = \sum_{i=1}^{\ell} y_i \alpha_i^* \mathbf{x}_i$ *realises the maximal margin hyperplane with geometric margin*

$$\gamma = 1 / \left\| \mathbf{w}^* \right\|_2 .$$

Remark 6.4 The value of b does not appear in the dual problem and so b^* must be found making use of the primal constraints:

$$b^* = -\frac{\max_{y_i=-1} \left(\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle \right) + \min_{y_i=1} \left(\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle \right)}{2}$$

Theorem 5.21 in Chapter 5 applies to this optimisation problem. The Karush–Kuhn–Tucker complementarity conditions provide useful information about the structure of the solution. The conditions state that the optimal solutions $\boldsymbol{\alpha}^*$, $(\mathbf{w}^*, b^*)$ must satisfy

$$\alpha_i^* \left[y_i \left(\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + b^* \right) - 1 \right] = 0, \qquad i = 1, \dots, \ell.$$

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

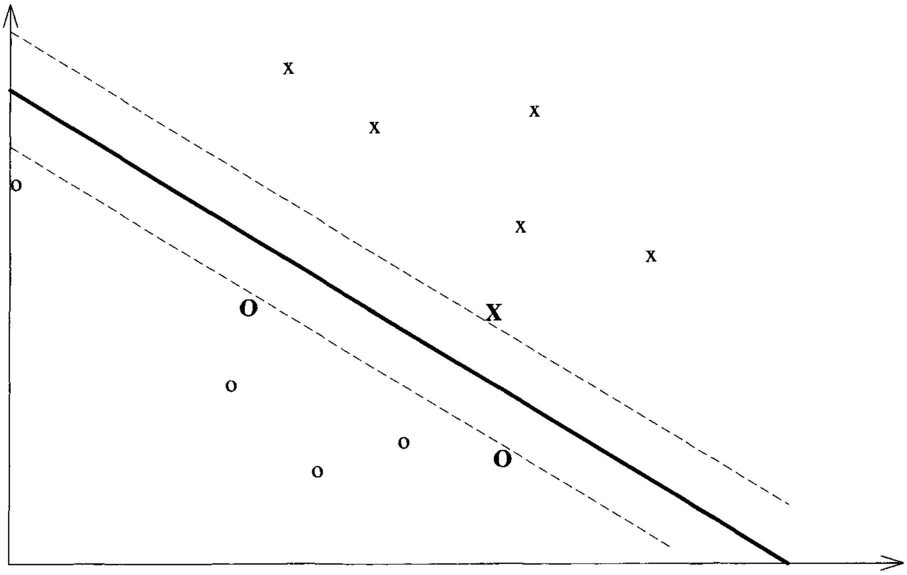

Figure 6.1: A maximal margin hyperplane with its support vectors highlighted

This implies that only for inputs $\mathbf{x}_i$ for which the functional margin is one and that therefore lie closest to the hyperplane are the corresponding α_i^* non-zero. All the other parameters α_i^* are zero. Hence, in the expression for the weight vector only these points are involved. It is for this reason that they are called *support vectors*, see Figure 6.1 We will denote the set of indices of the support vectors with sv.

Furthermore the optimal hyperplane can be expressed in the dual representation in terms of this subset of the parameters:

$$\begin{aligned} f(\mathbf{x},\boldsymbol{\alpha}^*,b^*) &= \sum_{i=1}^{\ell} y_i\alpha_i^* \langle \mathbf{x}_i \cdot \mathbf{x}\rangle + b^* \\ &= \sum_{i\in\mathrm{sv}} y_i\alpha_i^* \langle \mathbf{x}_i \cdot \mathbf{x}\rangle + b^*. \end{aligned}$$

The Lagrange multipliers associated with each point become the dual variables, giving them an intuitive interpretation quantifying how important a given training point is in forming the final solution. Points that are not support vectors have no influence, so that in non-degenerate cases slight perturbations of such points will not affect the solution. A similar meaning was found in the case of the dual representations for the perceptron learning algorithm, where the dual variable was proportional to the number of mistakes made by the hypothesis on a given point during the training.

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

Another important consequence of the Karush–Kuhn–Tucker complementarity conditions is that for $j \in \mathrm{sv}$,

$$y_j f(\mathbf{x}_j, \boldsymbol{\alpha}^*, b^*) = y_j \left(\sum_{i \in \mathrm{sv}} y_i \alpha_i^* \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle + b^* \right) = 1,$$

and therefore

$$\begin{aligned} \langle \mathbf{w}^* \cdot \mathbf{w}^* \rangle &= \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \\ &= \sum_{j \in \mathrm{sv}} \alpha_j^* y_j \sum_{i \in \mathrm{sv}} y_i \alpha_i^* \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \\ &= \sum_{j \in \mathrm{sv}} \alpha_j^* \left(1 - y_j b^* \right) \\ &= \sum_{i \in \mathrm{sv}} \alpha_i^*. \end{aligned}$$

We therefore have the following proposition.

Proposition 6.5 *Consider a linearly separable training sample*

$$S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell)),$$

and suppose the parameters $\boldsymbol{\alpha}^*$ *and* b^* *solve the dual optimisation problem (6.1). Then the weight vector* $\mathbf{w} = \sum_{i=1}^{\ell} y_i \alpha_i^* \mathbf{x}_i$ *realises the maximal margin hyperplane with geometric margin*

$$\gamma = 1/\left\| \mathbf{w} \right\|_2 = \left(\sum_{i \in \mathrm{sv}} \alpha_i^* \right)^{-1/2}.$$

Both the dual objective and the decision function have the remarkable property that the data only appear inside an inner product. This will make it possible to find and use optimal hyperplanes in the feature space through the use of kernels as shown in the following proposition.

Proposition 6.6 *Consider a training sample*

$$S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell))$$

that is linearly separable in the feature space implicitly defined by the kernel $K(\mathbf{x}, \mathbf{z})$ *and suppose the parameters* $\boldsymbol{\alpha}^*$ *and* b^* *solve the following quadratic optimisation problem:*

$$\begin{array}{ll} \textit{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j), \\ \textit{subject to} & \sum_{i=1}^{\ell} y_i \alpha_i = 0, \\ & \alpha_i \geq 0,\ i = 1, \dots, \ell. \end{array} \tag{6.2}$$

Then the decision rule given by $\operatorname{sgn}(f(\mathbf{x}))$, *where* $f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$, *is equivalent to the maximal margin hyperplane in the feature space implicitly defined by the kernel* $K(\mathbf{x}, \mathbf{z})$ *and that hyperplane has geometric margin*

$$\gamma = \left(\sum_{i \in \mathrm{sv}} \alpha_i^* \right)^{-1/2}.$$

Note that the requirement that the kernel satisfy Mercer's conditions is equivalent to the requirement that the matrix with entries $\left(K(\mathbf{x}_i, \mathbf{x}_j)\right)_{i,j=1}^{\ell}$ be positive definite for all training sets. This in turn means that the optimisation problem (6.2) is convex since the matrix $\left(y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)\right)_{i,j=1}^{\ell}$ is also positive definite. Hence, the property required for a kernel function to define a feature space also ensures that the maximal margin optimisation problem has a unique solution that can be found efficiently. This rules out the problem of local minima encountered in training neural networks.

Remark 6.7 The maximal margin classifier is motivated by Theorem 4.18 which bounds the generalisation error in terms of the margin and the radius of a ball centred at the origin containing the data. One advantage of motivating an algorithm using such a theorem is that we can compute a bound on the generalisation as an output of the learning algorithm. The value of the margin is given in Proposition 6.6, while the radius of the ball centred at the origin in feature space can be computed as

$$R = \max_{1 \le i \le \ell} \left(K(\mathbf{x}_i, \mathbf{x}_i) \right).$$

Unfortunately, though the strategy suggested by the theorem has been shown to be very effective, the constants involved typically make the actual value of the resulting bound unrealistic. There is still, however, the potential for using the bounds to choose between for example different kernels, where it is the relative size that is important, though the accuracy of the bounds is still the subject of active research.

An important result from the optimisation theory chapter is Theorem 5.15 stating that the primal objective is always bigger than the dual objective. Since the problem we are considering satisfies the conditions of Theorem 5.20, there is no duality gap at the optimal solution. We can therefore use any difference between the primal and dual values as an indicator of convergence. We will call this difference the feasibility gap. Let $\hat{\alpha}$ be the current value of the dual variables. The weight vector is calculated from setting the derivative of the Lagrangian equal to zero, and so the current value of the weight vector $\hat{\mathbf{w}}$ is the one which minimises $L(\mathbf{w}, b, \hat{\boldsymbol{\alpha}})$ for the given $\hat{\boldsymbol{\alpha}}$. Hence, this difference can be computed as follows:

$$\begin{aligned}
W(\hat{\boldsymbol{\alpha}}) - \frac{1}{2}\left\|\hat{\mathbf{w}}\right\|^2 &= \inf_{\mathbf{w},b} L(\mathbf{w}, b, \hat{\boldsymbol{\alpha}}) - \frac{1}{2}\left\|\hat{\mathbf{w}}\right\|^2 \\
&= L(\hat{\mathbf{w}}, b, \hat{\boldsymbol{\alpha}}) - \frac{1}{2}\left\|\hat{\mathbf{w}}\right\|^2 \\
&= -\sum_{i=1}^{\ell} \hat{\alpha}_i \left[y_i \left(\langle \hat{\mathbf{w}} \cdot \mathbf{x}_i \rangle + b \right) - 1 \right] \\
&= \sum_{i=1}^{\ell} \hat{\alpha}_i - \sum_{i,j=1}^{\ell} \hat{\alpha}_i y_i y_j \hat{\alpha}_j \langle \mathbf{x}_j \cdot \mathbf{x}_i \rangle,
\end{aligned}$$

which is minus the sum of the Karush–Kuhn–Tucker complementarity conditions. Note that this will correspond to the difference between primal and dual feasible solutions provided $\hat{\mathbf{w}}$ satisfies the primal constraints, that is provided $y_i \left(\langle \hat{\mathbf{w}} \cdot \mathbf{x}_i \rangle + b \right) \geq 1$ for all i, which is equivalent to

$$y_i \left(\sum_{j=1}^{\ell} y_j \hat{\alpha}_j \langle \mathbf{x}_j \cdot \mathbf{x}_i \rangle + b \right) \geq 1.$$

There is no guarantee that this will hold, and so computation of a feasibility gap is not straightforward in the maximal margin case. We will see below that for one of the soft margin cases we can estimate the feasibility gap.

The fact that only a subset of the Lagrange multipliers is non-zero is referred to as *sparseness*, and means that the support vectors contain all the information necessary to reconstruct the hyperplane. Even if all of the other points were removed the same maximal separating hyperplane would be found for the remaining subset of the support vectors. This can also be seen from the dual problem, since removing rows and columns corresponding to non-support vectors leaves the same optimisation problem for the remaining submatrix. Hence, the optimal solution remains unchanged. This shows that the maximal margin hyperplane is a compression scheme according to the definition of Section 4.4, since given the subset of support vectors we can reconstruct the maximal margin hyperplane that will correctly classify the whole training set. Applying Theorem 4.25, we obtain the following result.

Theorem 6.8 *Consider thresholding real-valued linear functions $\mathscr{L}$ with unit weight vectors on an inner product space X. For any probability distribution $\mathscr{D}$ on $X \times \{-1, 1\}$, with probability $1 - \delta$ over ℓ random examples S, the maximal margin hyperplane has error no more than*

$$\operatorname{err}_{\mathscr{D}}(f) \leq \frac{1}{\ell - d} \left(d \log \frac{e\ell}{d} + \log \frac{\ell}{\delta} \right),$$

where $d = \#\mathrm{sv}$ is the number of support vectors.

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

The theorem shows that the fewer the number of support vectors the better generalisation can be expected. This is closely related to the Ockham approach of finding a compact representation of the classification function. The nice property of the bound is that it does not depend explicitly on the dimension of the feature space.

Remark 6.9 A slightly tighter bound on the *expected* generalisation error in terms of the same quantities can be obtained by a leave-one-out argument. Since, when a non-support vector is omitted, it is correctly classified by the remaining subset of the training data the leave-one-out estimate of the generalisation error is

$$\frac{\# \text{sv}}{\ell}.$$

A cyclic permutation of the training set shows that the expected error of a test point is bounded by this quantity. The use of an expected generalisation bound gives no guarantee about its variance and hence its reliability. Indeed leave-one-out bounds are known to suffer from this problem. Theorem 6.8 can be seen as showing that in the case of maximal margin classifiers an only very slightly weaker bound does hold with high probability and hence that in this case the variance cannot be too high.

The maximal margin classifier does not attempt to control the number of support vectors and yet in practice there are frequently very few support vectors. This sparseness of the solution will also motivate a number of implementation techniques for dealing with large datasets, which we will discuss in more detail in Chapter 7.

The only degree of freedom in the maximal margin algorithm is the choice of kernel, which amounts to model selection. Any prior knowledge we have of the problem can help in choosing a parametrised kernel family, and then model selection is reduced to adjusting the parameters. For most classes of kernels, for example polynomial or Gaussian, it is always possible to find a kernel parameter for which the data become separable. In general, however, forcing separation of the data can easily lead to overfitting, particularly when noise is present in the data.

In this case, outliers would typically be characterised by a large Lagrange multiplier, and the procedure could be used for data cleaning, since it can rank the training data according to how difficult they are to classify correctly.

This algorithm provides the starting point for the many variations on this theme proposed in the last few years and attempting to address some of its weaknesses: that it is sensitive to the presence of noise; that it only considers two classes; that it is not expressly designed to achieve sparse solutions.

Remark 6.10 Note that in SVMs the margin has two effects. On the one hand, its maximisation ensures low fat-shattering dimension, and hence better generalisation, while on the other hand the margin is the origin of the sparseness of the solution vector, as the inequality constraints generate the Karush–Kuhn–Tucker complementarity conditions.

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

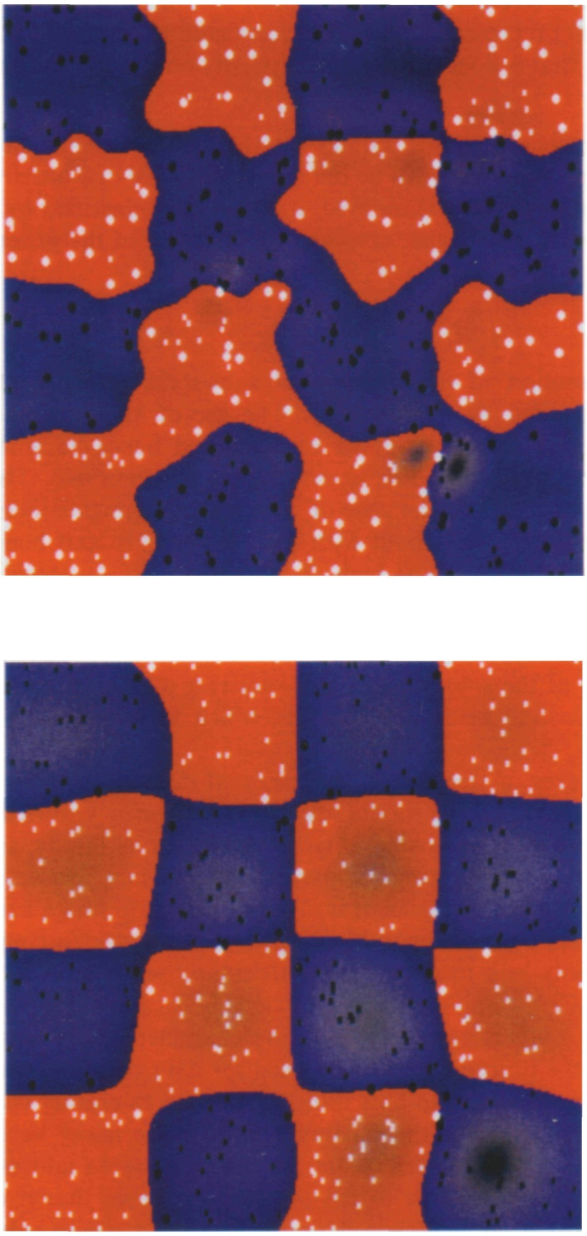

Figure 6.2: For caption see facing page

6.1.2 Soft Margin Optimisation

The maximal margin classifier is an important concept, as a starting point for the analysis and construction of more sophisticated Support Vector Machines, but it cannot be used in many real-world problems (we will see an exception in Chapter 8): if the data are noisy, there will in general be no linear separation in the feature space (unless we are ready to use very powerful kernels, and hence overfit the data). The main problem with the maximal margin classifier is that it always produces perfectly a consistent hypothesis, that is a hypothesis with no training error. This is of course a result of its motivation in terms of a bound that depends on the margin, a quantity that is negative unless the data are perfectly separated.

The dependence on a quantity like the margin opens the system up to the danger of falling hostage to the idiosyncrasies of a few points. In real data, where noise can always be present, this can result in a brittle estimator. Furthermore, in the cases where the data are not linearly separable in the feature space, the optimisation problem cannot be solved as the primal has an empty feasible region and the dual an unbounded objective function. These problems motivate using the more robust measures of the *margin distribution* introduced in Chapter 4. Such measures can tolerate noise and outliers, and take into consideration the positions of more training points than just those closest to the boundary.

Recall that the primal optimisation problem for the maximal margin case is the following:

$$\begin{array}{ll} \text{minimise}_{\mathbf{w},b} & \langle \mathbf{w} \cdot \mathbf{w} \rangle, \\ \text{subject to} & y_i \left(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \right) \geq 1, i = 1, \ldots, \ell. \end{array}$$

In order to optimise the margin slack vector we need to introduce slack variables

Figure 6.2: The figures show the result of the maximum margin SVM for learning a chess board from points generated according to the uniform distribution using Gaussian kernels with different values of σ. The white dots are the positive points and the black dots the negative ones. The support vectors are indicated by large dots. The red area comprises those points that are positively classified by the decision function, while the area classified negative is coloured blue. Notice that in both cases the classification of the training set is consistent. The size of the functional margin is indicated by the level of shading. The images make clear how the accuracy of the resulting classifier can be affected by the choice of kernel parameter. In image (b) with the large value of σ, each region has only a small number of support vectors and the darker shading clearly indicates where the machine has more confidence of its classification. In contast image (a) has a more complex boundary, significantly more support vectors, and there are very few regions with darker shading

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

to allow the margin constraints to be violated

$$\begin{array}{ll}\text{subject to} & y_i\left(\langle \mathbf{w} \cdot \mathbf{x}_i\rangle + b\right) \geq 1 - \xi_i,\ i = 1,\dots,\ell, \\ & \xi_i \geq 0,\ i = 1,\dots,\ell.\end{array}$$

Theorem 4.22 in Chapter 4 bounds the generalisation error in terms of the 2-norm of the margin slack vector, the so-called 2-norm soft margin, which contains the ξ_i scaled by the norm of the weight vector $\mathbf{w}$. Hence, the equivalent expression on which the generalisation depends is

$$\begin{aligned}\frac{R^2 + \frac{\|\xi\|_2^2}{\|\mathbf{w}\|_2^2}}{\gamma^2} &= \|\mathbf{w}\|_2^2\left(R^2 + \frac{\|\xi\|_2^2}{\|\mathbf{w}\|_2^2}\right) \\ &= \|\mathbf{w}\|_2^2 R^2 + \|\xi\|_2^2,\end{aligned}$$

suggesting that an optimal choice for C in the objective function of the resulting optimisation problem should be R^{-2}:

$$\begin{array}{ll}\text{minimise}_{\xi,\mathbf{w},b} & \langle \mathbf{w} \cdot \mathbf{w}\rangle + C\sum_{i=1}^{\ell}\xi_i^2 \\ \text{subject to} & y_i\left(\langle \mathbf{w} \cdot \mathbf{x}_i\rangle + b\right) \geq 1 - \xi_i,\ i = 1,\dots,\ell, \\ & \xi_i \geq 0,\ i = 1,\dots,\ell.\end{array} \qquad (6.3)$$

Notice that if $\xi_i < 0$, then the first constraint will still hold if we set $\xi_i = 0$, while this change will reduce the value of the objective function. Hence, the optimal solution for the problem obtained by removing the positivity constraint on ξ_i will coincide with the optimal solution of equation (6.3). Hence we obtain the solution to equation (6.3) by solving the following optimisation problem:

$$\begin{array}{ll}\text{minimise}_{\xi,\mathbf{w},b} & \langle \mathbf{w} \cdot \mathbf{w}\rangle + C\sum_{i=1}^{\ell}\xi_i^2, \\ \text{subject to} & y_i\left(\langle \mathbf{w} \cdot \mathbf{x}_i\rangle + b\right) \geq 1 - \xi_i,\ i = 1,\dots,\ell,\end{array} \qquad (6.4)$$

In practice the parameter C is varied through a wide range of values and the optimal performance assessed using a separate validation set or a technique known as cross-validation for verifying performance using only the training set. As the parameter C runs through a range of values, the norm $\|\mathbf{w}\|_2$ varies smoothly through a corresponding range. Hence, for a particular problem, choosing a particular value for C corresponds to choosing a value for $\|\mathbf{w}\|_2$, and then minimising $\|\xi\|_2$ for that size of $\mathbf{w}$. This approach is also adopted in the 1-norm case where the optimisation problem minimises a combination of the norm of the weights and the 1-norm of the slack variables that does not exactly match that found in Theorem 4.24:

$$\begin{array}{ll}\text{minimise}_{\xi,\mathbf{w},b} & \langle \mathbf{w} \cdot \mathbf{w}\rangle + C\sum_{i=1}^{\ell}\xi_i, \\ \text{subject to} & y_i\left(\langle \mathbf{w} \cdot \mathbf{x}_i\rangle + b\right) \geq 1 - \xi_i,\ i = 1,\dots,\ell, \\ & \xi_i \geq 0,\ i = 1,\dots,\ell.\end{array} \qquad (6.5)$$

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

Since there is a value of C corresponding to the optimal choice of $\|\mathbf{w}\|_2$, that value of C will give the optimal bound as it will correspond to finding the minimum of $\|\xi\|_1$ with the given value for $\|\mathbf{w}\|_2$.

We will devote the next two subsubsections to investigating the duals of the two margin slack vector problems creating the so-called soft margin algorithms.

2-Norm Soft Margin – Weighting the Diagonal

The primal Lagrangian for the problem of equation (6.4) is

$$L(\mathbf{w}, b, \xi, \alpha) = \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle + \frac{C}{2}\sum_{i=1}^{\ell} \xi_i^2 - \sum_{i=1}^{\ell} \alpha_i \left[y_i \left(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \right) - 1 + \xi_i \right]$$

where $\alpha_i \geq 0$ are the Lagrange multipliers, as described in Chapter 5.

The corresponding dual is found by differentiating with respect to $\mathbf{w}$, ξ and b, imposing stationarity,

$$\begin{aligned} \frac{\partial L(\mathbf{w}, b, \xi, \alpha)}{\partial \mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{\ell} y_i \alpha_i \mathbf{x}_i = \mathbf{0}, \\ \frac{\partial L(\mathbf{w}, b, \xi, \alpha)}{\partial \xi} &= C\xi - \alpha = \mathbf{0}, \\ \frac{\partial L(\mathbf{w}, b, \xi, \alpha)}{\partial b} &= \sum_{i=1}^{\ell} y_i \alpha_i = 0, \end{aligned}$$

and resubstituting the relations obtained into the primal to obtain the following adaptation of the dual objective function:

$$\begin{aligned} L(\mathbf{w}, b, \xi, \alpha) &= \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle + \frac{1}{2C}\langle \alpha \cdot \alpha \rangle - \frac{1}{C}\langle \alpha \cdot \alpha \rangle \\ &= \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle - \frac{1}{2C}\langle \alpha \cdot \alpha \rangle . \end{aligned}$$

Hence, maximising the above objective over α is equivalent to maximising

$$W(\alpha) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \left(\langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle + \frac{1}{C}\delta_{ij} \right),$$

where δ_{ij} is the Kronecker δ defined to be 1 if $i = j$ and 0. The corresponding Karush–Kuhn–Tucker complementarity conditions are

$$\alpha_i \left[y_i(\langle \mathbf{x}_i \cdot \mathbf{w} \rangle + b) - 1 + \xi_i \right] = 0, \quad i = 1, \ldots, \ell.$$

Hence, we have the following result in which we have moved directly to the more general kernel version.

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

Proposition 6.11 *Consider classifying a training sample*

$$S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell)),$$

using the feature space implicitly defined by the kernel $K(\mathbf{x}, \mathbf{z})$, and suppose the parameters $\boldsymbol{\alpha}^*$ solve the following quadratic optimisation problem:

$$\begin{array}{ll} \text{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j \left(K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{C} \delta_{ij} \right), \\ \text{subject to} & \sum_{i=1}^{\ell} y_i \alpha_i = 0, \\ & \alpha_i \geq 0, \ i = 1, \dots, \ell. \end{array}$$

Let $f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$, where b^* is chosen so that $y_i f(\mathbf{x}_i) = 1 - \alpha_i^*/C$ for any i with $\alpha_i^* \neq 0$. Then the decision rule given by $\operatorname{sgn}(f(\mathbf{x}))$ is equivalent to the hyperplane in the feature space implicitly defined by the kernel $K(\mathbf{x}, \mathbf{z})$ which solves the optimisation problem (6.3), where the slack variables are defined relative to the geometric margin

$$\gamma = \left(\sum_{i \in \text{sv}} \alpha_i^* - \frac{1}{C} \langle \boldsymbol{\alpha}^* \cdot \boldsymbol{\alpha}^* \rangle \right)^{-1/2}.$$

Proof The value of b^* is chosen using the relation $\alpha_i = C\xi_i$ and by reference to the primal constraints which by the Karush–Kuhn–Tucker complementarity conditions

$$\alpha_i \left[y_i \left(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \right) - 1 + \xi_i \right] = 0, \ i = 1, \dots, \ell,$$

must be equalities for non-zero α_i. It remains to compute the norm of $\mathbf{w}^*$ which defines the size of the geometric margin.

$$\begin{aligned} \langle \mathbf{w}^* \cdot \mathbf{w}^* \rangle &= \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* K(\mathbf{x}_i, \mathbf{x}_j) \\ &= \sum_{j \in \text{sv}} \alpha_j^* y_j \sum_{i \in \text{sv}} y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{x}_j) \\ &= \sum_{j \in \text{sv}} \alpha_j^* \left(1 - \xi_j^* - y_j b^* \right) \\ &= \sum_{i \in \text{sv}} \alpha_i^* - \sum_{i \in \text{sv}} \alpha_i^* \xi_i^* \\ &= \sum_{i \in \text{sv}} \alpha_i^* - \frac{1}{C} \langle \boldsymbol{\alpha}^* \cdot \boldsymbol{\alpha}^* \rangle. \end{aligned}$$

□

This is still a quadratic programming problem, and can be used with the same methods as used for the maximal margin hyperplane. The only change is the addition of $1/C$ to the diagonal of the inner product matrix associated with the training set. This has the effect of adding $1/C$ to the eigenvalues of the matrix, rendering the problem better conditioned. We can therefore view the new problem as simply a change of kernel

$$K'(\mathbf{x},\mathbf{z}) = K(\mathbf{x},\mathbf{z}) + \frac{1}{C}\delta_{\mathbf{x}}(\mathbf{z}).$$

1-Norm Soft Margin – the Box Constraint

The corresponding Lagrangian for the 1-norm soft margin optimisation problem is

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

with $\alpha_i \geq 0$ and $r_i \geq 0$. The corresponding dual is found by differentiating with respect to $\mathbf{w}$, $\boldsymbol{\xi}$ and b, imposing stationarity,

$$\begin{aligned} \frac{\partial L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\mathbf{r})}{\partial\mathbf{w}} &= \mathbf{w} - \sum_{i=1}^{\ell} y_i\alpha_i\mathbf{x}_i = \mathbf{0}, \\ \frac{\partial L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\mathbf{r})}{\partial\xi_i} &= C-\alpha_i-r_i = 0, \\ \frac{\partial L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\mathbf{r})}{\partial b} &= \sum_{i=1}^{\ell} y_i\alpha_i = 0, \end{aligned}$$

and resubstituting the relations obtained into the primal; we obtain the following adaptation of the dual objective function:

$$L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\mathbf{r}) = \sum_{i=1}^{\ell}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} y_iy_j\alpha_i\alpha_j\langle\mathbf{x}_i\cdot\mathbf{x}_j\rangle,$$

which curiously is identical to that for the maximal margin. The only difference is that the constraint $C-\alpha_i-r_i=0$, together with $r_i \geq 0$, enforces $\alpha_i \leq C$, while $\xi_i \neq 0$ only if $r_i = 0$ and therefore $\alpha_i = C$. The Karush–Kuhn–Tucker complementarity conditions are therefore

$$\begin{aligned} &\alpha_i\left[y_i(\langle\mathbf{x}_i\cdot\mathbf{w}\rangle + b) - 1 + \xi_i\right] = 0, && i = 1,\dots,\ell, \\ &\xi_i(\alpha_i - C) = 0, && i = 1,\dots,\ell. \end{aligned}$$

Notice that the KKT conditions implies that non-zero slack variables can only occur when $\alpha_i = C$. The points with non-zero slack variables are $1/\left\|\mathbf{w}\right\|$-margin errors, as their geometric margin is less than $1/\left\|\mathbf{w}\right\|$. Points for which $0 < \alpha_i < C$ lie at the target distance of $1/\left\|\mathbf{w}\right\|$ from the hyperplane. We therefore have the following proposition.

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

Proposition 6.12 *Consider classifying a training sample*

$$S = ((\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)),$$

using the feature space implicitly defined by the kernel $K(\mathbf{x}, \mathbf{z})$, *and suppose the parameters* $\boldsymbol{\alpha}^*$ *solve the following quadratic optimisation problem:*

$$\begin{array}{ll} \textit{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j), \\ \textit{subject to} & \sum_{i=1}^{\ell} y_i \alpha_i = 0, \\ & C \geq \alpha_i \geq 0,\ i = 1, \ldots, \ell. \end{array} \tag{6.6}$$

Let $f(\mathbf{x}) = \sum_{i=1}^{\ell} y_i \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$, *where* b^* *is chosen so that* $y_i f(\mathbf{x}_i) = 1$ *for any* i *with* $C > \alpha_i^* > 0$. *Then the decision rule given by* $\operatorname{sgn}(f(\mathbf{x}))$ *is equivalent to the hyperplane in the feature space implicitly defined by the kernel* $K(\mathbf{x}, \mathbf{z})$ *that solves the optimisation problem (6.5), where the slack variables are defined relative to the geometric margin*

$$\gamma = \left(\sum_{i,j \in \mathrm{sv}} y_i y_j \alpha_i^* \alpha_j^* K(\mathbf{x}_i, \mathbf{x}_j) \right)^{-1/2}.$$

Proof The value of b^* is chosen using the Karush–Kuhn–Tucker complementarity conditions which imply that if $C > \alpha_i^* > 0$ both $\xi_i^* = 0$ and

$$y_i(\langle \mathbf{x}_i \cdot \mathbf{w}^* \rangle + b^*) - 1 + \xi_i^* = 0.$$

The norm of $\mathbf{w}^*$ is clearly given by the expression

$$\begin{aligned} \langle \mathbf{w}^* \cdot \mathbf{w}^* \rangle &= \sum_{i,j=1}^{\ell} y_i y_j \alpha_i^* \alpha_j^* K(\mathbf{x}_i, \mathbf{x}_j) \\ &= \sum_{j \in \mathrm{sv}} \sum_{i \in \mathrm{sv}} y_i y_j \alpha_i^* \alpha_j^* K(\mathbf{x}_i, \mathbf{x}_j). \end{aligned}$$

□

So surprisingly this problem is equivalent to the maximal margin hyperplane, with the additional constraint that all the α_i are upper bounded by C. This gives rise to the name *box constraint* that is frequently used to refer to this formulation, since the vector $\boldsymbol{\alpha}$ is constrained to lie inside the box with side length C in the positive orthant. The trade-off parameter between accuracy and regularisation directly controls the size of the α_i. This makes sense intuitively as the box constraints limit the influence of outliers, which would otherwise have large Lagrange multipliers. The constraint also ensures that the feasible region is bounded and hence that the primal always has a non-empty feasible region.

Remark 6.13 One problem with the soft margin approach suggested is the choice of parameter C. Typically a range of values must be tried before the best choice for a particular training set can be selected. Furthermore the scale of the

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

parameter is affected by the choice of feature space. It has been shown, however, that the solutions obtained for different values of C in the optimisation problem (6.6) are the same as those obtained as v is varied between 0 and 1 in the optimisation problem

$$\begin{array}{ll} \text{maximise} & W(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ \text{subj. to} & \sum_{i=1}^{\ell} y_i \alpha_i = 0, \\ & \sum_{i=1}^{\ell} \alpha_i \geq v \\ & 1/\ell \geq \alpha_i \geq 0,\ i = 1, \dots, \ell. \end{array}$$

In this parametrisation v places a lower bound on the sum of the α_i, which causes the linear term to be dropped from the objective function. It can be shown that the proportion of the training set that are margin errors is upper bounded by v, while v provides a lower bound on the total number of support vectors. Therefore v gives a more transparent parametrisation of the problem which does not depend on the scaling of the feature space, but only on the noise level in the data. For details of this approach and its application to regression see pointers in Section 6.5.

In the case of the 1-norm margin slack vector optimisation the feasibility gap can be computed since the ξ_i are not specified when moving to the dual and so can be chosen to ensure that the primary problem is feasible, by taking

$$\xi_i = \max\left(0, 1 - y_i\left(\sum_{j=1}^{\ell} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b\right)\right),$$

where $\boldsymbol{\alpha}$ is the current estimate for the dual problem and b has been chosen so that $y_i f(\mathbf{x}_i) = 1$ for some i with $C > \alpha_i > 0$. Once the primal problem is feasible the gap between the value of the primal and dual objectives becomes the sum of the Karush–Kuhn–Tucker complementarity conditions by the construction of the Lagrangian:

$$-L(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\alpha}, \mathbf{r}) + \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle + C\sum_{i=1}^{\ell} \xi_i =$$

$$= \sum_{i=1}^{\ell} \alpha_i \left[y_i \left(\sum_{j=1}^{\ell} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) + b \right) - 1 + \xi_i \right] + \sum_{i=1}^{\ell} r_i \xi_i,$$

where $r_i = C - \alpha_i$. Hence, using the constraint on $\boldsymbol{\alpha}$, the difference between primal and dual objectives is given by

$$
\begin{aligned}
\sum_{i=1}^{\ell} \alpha_i \left[y_i(\langle \mathbf{x}_i \cdot \mathbf{w} \rangle + b) - 1 + \xi_i \right] + \sum_{i=1}^{\ell} r_i \xi_i &= \\
&= \sum_{i=1}^{\ell} \alpha_i \left[y_i \left(\sum_{j=1}^{\ell} y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) \right) - 1 \right] + C \sum_{i=1}^{\ell} \xi_i \\
&= \sum_{i,j=1}^{\ell} \alpha_i y_i y_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) - \sum_{i=1}^{\ell} \alpha_i + C \sum_{i=1}^{\ell} \xi_i \\
&= \sum_{i=1}^{\ell} \alpha_i - 2W(\boldsymbol{\alpha}) + C \sum_{i=1}^{\ell} \xi_i .
\end{aligned}
$$

Remark 6.14 This explains why we noted that the maximal (or hard) margin case is an important concept in the solution of more sophisticated versions of the machine: both the 1- and the 2-norm soft margin machines lead to optimisation problems that are solved by relating them to the maximal margin case.

Remark 6.15 Historically, the soft margin machines were introduced before their justification in terms of margin distribution generalisation bounds. For this reason the 1-norm was preferred as it appeared closer to the percentile error bound. The results show that both 1- and 2-norm bounds on generalisation exist. The approach that performs better in practice will depend on the data and may be influenced by the type of noise that has influenced it.

Remark 6.16 The techniques developed for two-class classification have been generalised to the case where there are several categories. References to relevant papers will be given at the end of the chapter in Section 6.5.

Figure 6.3: This figure shows the decision boundaries that arise when using a Gaussian kernel with a fixed value of σ in the three different machines: (a) the maximal margin SVM, (b) the 2-norm soft margin SVM, and (c) the 1-norm soft margin SVM. The data are an artificially created two dimensional set, the white points being positive examples and the black points negative: the larger sized points are the support vectors. The red area comprises those points that are positively classified by the decision function, while the area classified negative is coloured blue. The size of the functional margin is indicated by the level of shading. Notice that the hard margin correctly classifies the whole training set at the expense of a more complex decision boundary. The two soft margin approaches both give smoother decision boundaries by misclassifying two positive examples

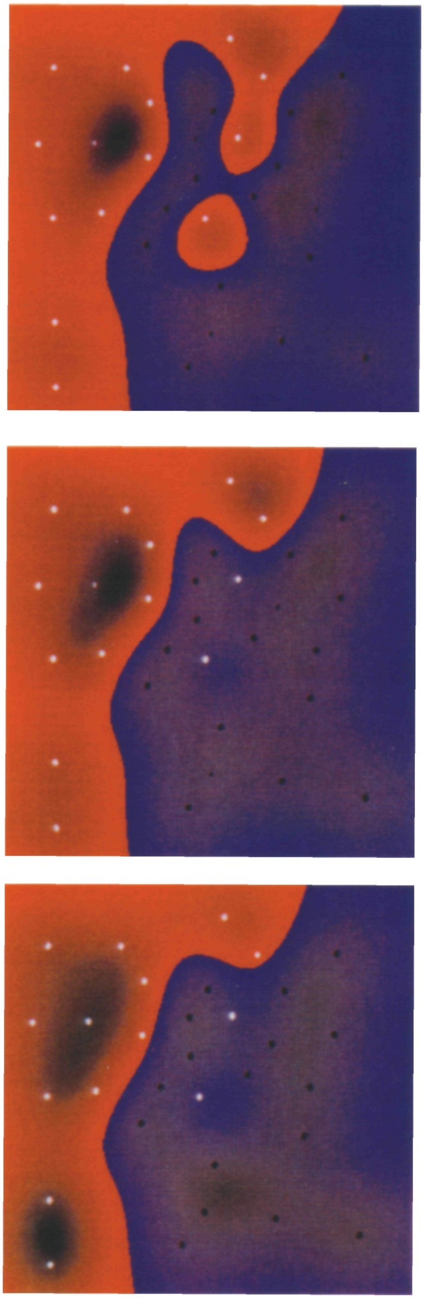

Figure 6.3: For caption see facing page

6.1.3 Linear Programming Support Vector Machines

Rather than using generalisation bounds based on margin distribution, one could try to enforce other learning biases, such as the sample compression bounds, as given in Theorems 4.25 and 6.8. This would lead for example to an algorithm for finding the sparsest separating hyperplane, regardless of its margin. The problem is computationally hard but can be approximated by minimising an estimate of the number of positive multipliers, $\sum_{i=1}^{\ell} \alpha_i$, while enforcing a margin of 1. Introducing slack variables in a way similar to that given above and working directly in the dual representation, one obtains the following linear optimisation problem:

$$\begin{array}{ll} \text{minimise} & L(\boldsymbol{\alpha}, \boldsymbol{\xi}) = \sum_{i=1}^{\ell} \alpha_i + C \sum_{i=1}^{\ell} \xi_i, \\ \text{subject to} & y_i \left[\sum_{j=1}^{\ell} \alpha_i \langle \mathbf{x}_i, \mathbf{x}_j \rangle + b \right] \geq 1 - \xi_i, \; i = 1, \dots, \ell, \\ & \alpha_i \geq 0, \; \xi_i \geq 0, \; i = 1, \dots, \ell. \end{array}$$

This type of approach was developed independently of the 2-norm maximal margin implicit in the definition of a standard Support Vector Machine. It has the advantage of relying on solving a linear programming problem as opposed to a convex quadratic one. The application of kernels to move to implicit feature spaces has also been made in this setting. Bounds on the generalisation directly in terms of $\sum_{i=1}^{\ell} \alpha_i$ have been derived more recently.

6.2 Support Vector Regression

The Support Vector method can also be applied to the case of regression, maintaining all the main features that characterise the maximal margin algorithm: a non-linear function is learned by a linear learning machine in a kernel-induced feature space while the capacity of the system is controlled by a parameter that does not depend on the dimensionality of the space. As in the classification case the learning algorithm minimises a convex functional and its solution is sparse.

As with the classification approach we motivate the approach by seeking to optimise the generalisation bounds given for regression in Chapter 4. These relied on defining a loss function that ignored errors that were within a certain distance of the true value. This type of function is referred to as an ε-insensitive loss function. Since this terminology is quite standard, we will risk using ε for this loss despite previously reserving this symbol for the generalisation error, that is the probability of misclassifying a randomly drawn test example.

Figure 6.4 shows an example of a one dimensional linear regression function with an ε-insensitive band. The variables ξ measure the cost of the errors on the training points. These are zero for all points inside the band. Figure 6.5 shows a similar situation for a non-linear regression function.

With many reasonable choices of loss function, the solution will be characterised as the minimum of a convex functional. Another motivation for considering the ε-insensitive loss function is that as with classification Support

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

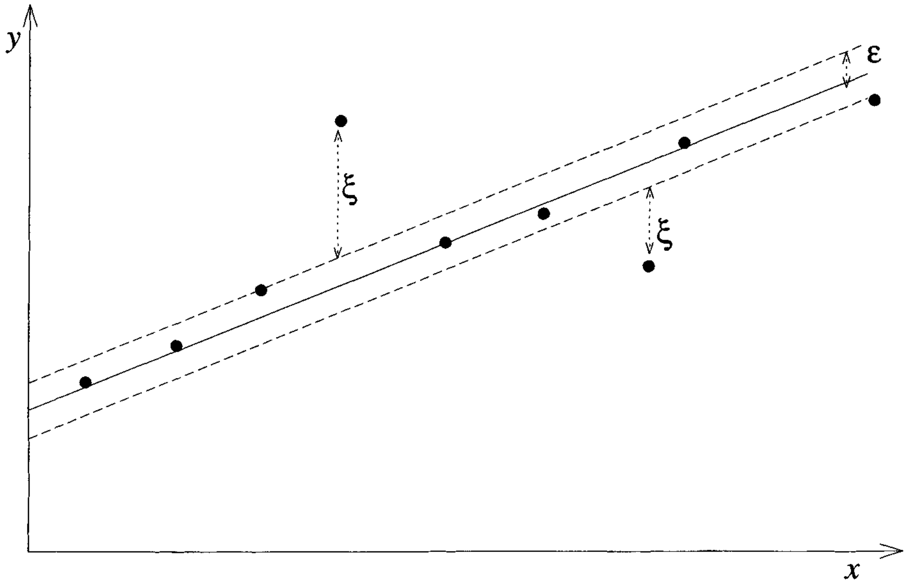

Figure 6.4: The insensitive band for a one dimensional linear regression problem

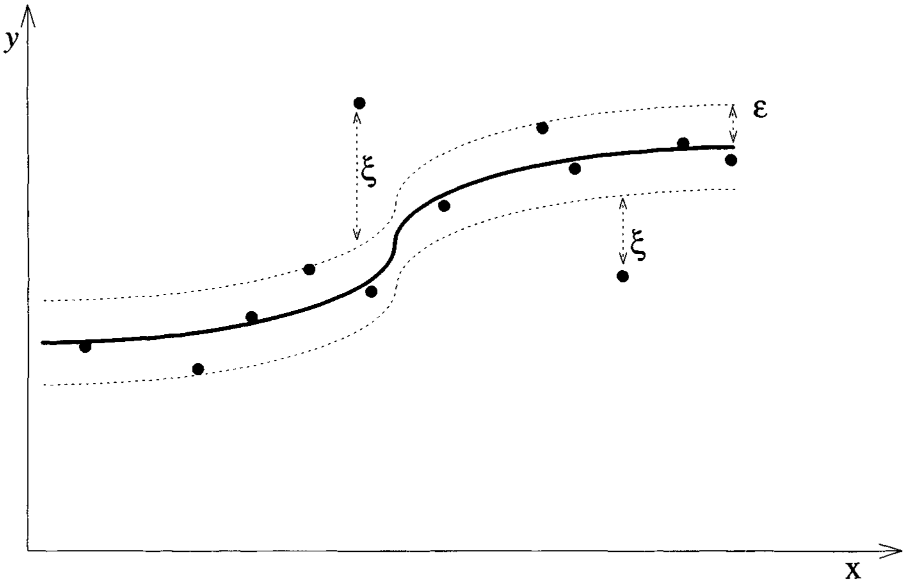

Figure 6.5: The insensitive band for a non-linear regression function

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

Vector Machines it will ensure sparseness of the dual variables. The idea of representing the solution by means of a small subset of training points has enormous computational advantages. Using the ε-insensitive loss function has that advantage, while still ensuring the existence of a global minimum and the optimisation of a reliable generalisation bound.

In this section we will first describe the ε-insensitive loss and then derive two algorithms from the bounds of Chapter 4, relating to the 1- or 2-norm of the loss vector. For comparison we will then give an algorithm for ridge regression in feature space, which does not enforce sparseness and hence presents more implementation problems. Finally, we will show how a popular regression algorithm based on Gaussian processes is equivalent to performing ridge regression in a feature space, and hence is intimately related to Support Vector Machines.

6.2.1 ε-Insensitive Loss Regression

Theorems 4.28 and 4.30 bound the generalisation performance of a linear regressor in terms of the norm of the weight vector and the 2- and 1-norms of the slack variables. The ε-insensitive loss function is equal to these slack variables.

Definition 6.17 The *(linear) ε-insensitive loss function* $L^{\varepsilon}(\mathbf{x}, y, f)$ is defined by

$$L^{\varepsilon}(\mathbf{x}, y, f) = |y - f(\mathbf{x})|_{\varepsilon} = \max\left(0, |y - f(\mathbf{x})| - \varepsilon\right),$$

where f is a real-valued function on a domain X, $\mathbf{x} \in \mathbf{X}$ and $y \in \mathbb{R}$. Similarly the *quadratic ε-insensitive loss* is given by

$$L_2^{\varepsilon}(\mathbf{x}, y, f) = |y - f(\mathbf{x})|_{\varepsilon}^2 .$$

If we compare this loss function with the margin slack vector defined in Definition 4.27 it is immediate that the margin slack variable $\xi\left((\mathbf{x}_i, y_i), f, \theta, \gamma\right)$ satisfies

$$\xi\left((\mathbf{x}_i, y_i), f, \theta, \gamma\right) = L^{\theta-\gamma}(\mathbf{x}_i, y_i, f).$$

Hence as indicated above the results of Chapter 4 use an ε-insensitive loss function with $\varepsilon = \theta - \gamma$. Figures 6.6 and 6.7 show the form of the linear and quadratic ε-insensitive losses for zero and non-zero ε as a function of $y - f(\mathbf{x})$.

Quadratic ε-Insensitive Loss

Theorem 4.28 suggests that we can optimise the generalisation of our regressor by minimising the sum of the quadratic ε-insensitive losses

$$R^2 \|\mathbf{w}\|^2 + \sum_{i=1}^{\ell} L_2^{\varepsilon}(\mathbf{x}_i, y_i, f),$$

where f is the function defined by the weight vector $\mathbf{w}$. Minimising this quantity has the advantage of minimising the bound for all values of γ, which implies

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

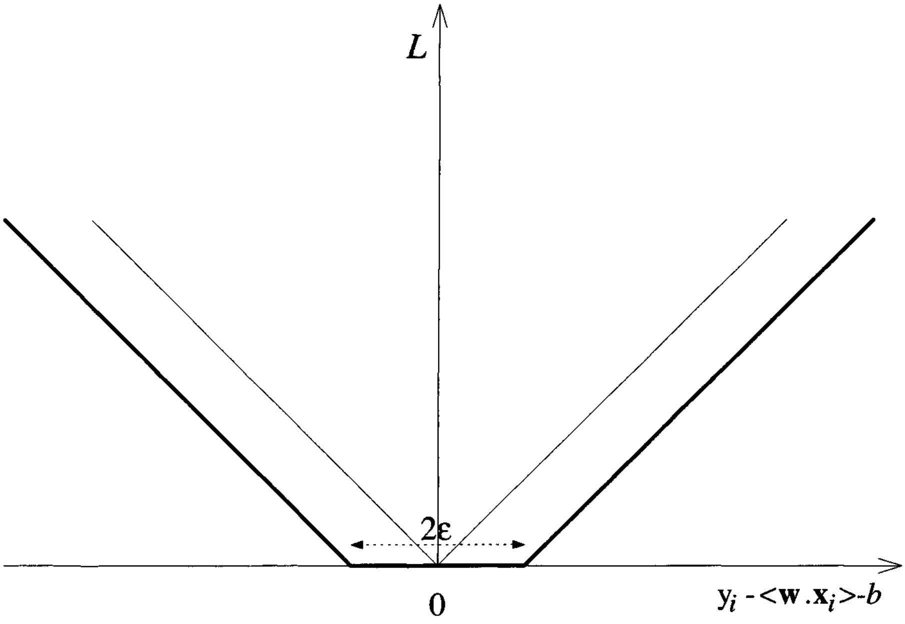

Figure 6.6: The linear ε-insensitive loss for zero and non-zero ε

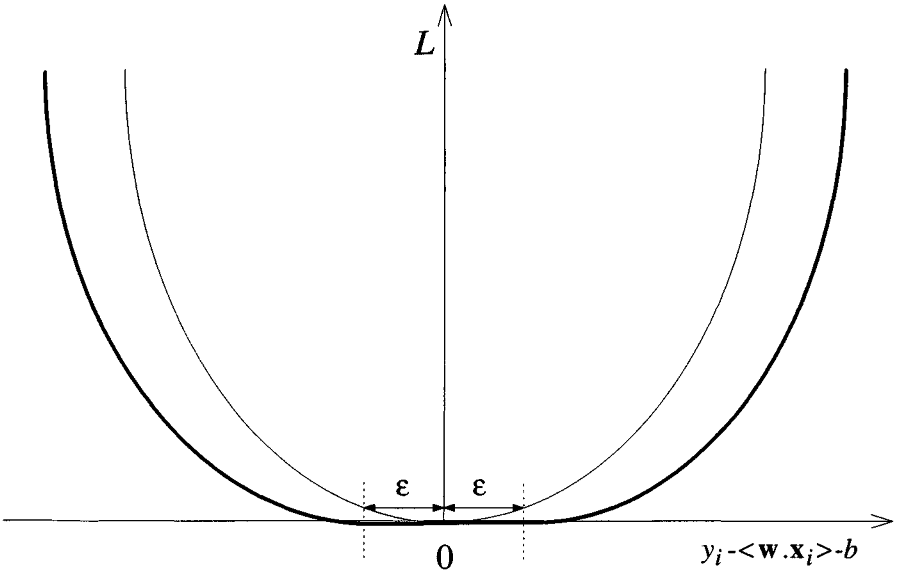

Figure 6.7: The quadratic ε-insensitive loss for zero and non-zero ε

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

that it is minimised for all values of $\theta = \varepsilon + \gamma$. As the classification case we introduce a parameter C to measure the trade-off between complexity and losses. The primal problem can therefore be defined as follows:

$$\begin{array}{ll} \text{minimise} & \|\mathbf{w}\|^2 + C\sum_{i=1}^{\ell}(\xi_i^2 + \hat{\xi}_i^2), \\ \text{subject to} & ((\mathbf{w}\cdot\mathbf{x}_i) + b) - y_i \le \varepsilon + \xi_i,\ i = 1,\dots,\ell, \\ & y_i - ((\mathbf{w}\cdot\mathbf{x}_i) + b) \le \varepsilon + \hat{\xi}_i,\ i = 1,\dots,\ell, \\ & \xi_i, \hat{\xi}_i \ge 0,\ i = 1,\dots,\ell, \end{array} \tag{6.7}$$

where we have introduced two slack variables, one for exceeding the target value by more than ε, and the other for being more than ε below the target. We will again typically consider solving this for a range of values of C and then use some validation method to select the optimal value of this parameter. The dual problem can be derived using the standard method and taking into account that $\xi_i\hat{\xi}_i = 0$ and therefore that the same relation $\alpha_i\hat{\alpha}_i = 0$ holds for the corresponding Lagrange multipliers:

$$\begin{array}{ll} \text{maximise} & \sum_{i=1}^{\ell} y_i(\hat{\alpha}_i - \alpha_i) - \varepsilon\sum_{i=1}^{\ell}(\hat{\alpha}_i + \alpha_i) \\ & \quad -\frac{1}{2}\sum_{i,j=1}^{\ell}(\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j)\left((\mathbf{x}_i\cdot\mathbf{x}_j) + \frac{1}{C}\delta_{ij}\right), \\ \text{subject to} & \sum_{i=1}^{\ell}(\hat{\alpha}_i - \alpha_i) = 0, \\ & \hat{\alpha}_i \ge 0,\ \alpha_i \ge 0,\ i = 1,\dots,\ell. \end{array}$$

The corresponding Karush–Kuhn–Tucker complementarity conditions are

$$\begin{array}{ll} \alpha_i\left((\mathbf{w}\cdot\mathbf{x}_i) + b - y_i - \varepsilon - \xi_i\right) = 0, & i = 1,\dots,\ell, \\ \hat{\alpha}_i\left(y_i - (\mathbf{w}\cdot\mathbf{x}_i) - b - \varepsilon - \hat{\xi}_i\right) = 0, & i = 1,\dots,\ell, \\ \xi_i\hat{\xi}_i = 0,\ \alpha_i\hat{\alpha}_i = 0, & i = 1,\dots,\ell, \end{array}$$

Remark 6.18 Note that by substituting $\boldsymbol{\beta} = \hat{\boldsymbol{\alpha}} - \boldsymbol{\alpha}$ and using the relation $\alpha_i\hat{\alpha}_i = 0$, it is possible to rewrite the dual problem in a way that more closely resembles the classification case.

$$\begin{array}{ll} \text{maximise} & \sum_{i=1}^{\ell} y_i\beta_i - \varepsilon\sum_{i=1}^{\ell}|\beta_i| - \frac{1}{2}\sum_{i,j=1}^{\ell}\beta_i\beta_j\left((\mathbf{x}_i\cdot\mathbf{x}_j) + \frac{1}{C}\delta_{ij}\right), \\ \text{subject to} & \sum_{i=1}^{\ell}\beta_i = 0,\ i = 1,\dots,\ell. \end{array}$$

For $y_i \in \{-1, 1\}$, the similarity becomes even more apparent when $\varepsilon = 0$ and if we use the variables $\hat{\beta}_i = y_i\beta_i$, the only difference being that $\hat{\beta}_i$ is not constrained to be positive unlike the corresponding α_i in the classification case. We will in fact use $\boldsymbol{\alpha}$ in place of $\boldsymbol{\beta}$ when we use this form later.

Remark 6.19 For non-zero ε the effect is to introduce an extra weight decay factor involving the dual parameters. The case $\varepsilon = 0$ corresponds to considering standard least squares linear regression with a weight decay factor controlled by the parameter C. As $C \to \infty$, the problem tends to an unconstrained least squares, which is equivalent to leaving the inner product matrix diagonal unchanged. Note that references to investigations of more general loss functions will be given at the end of the chapter.

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

Hence, we have the following result in which we have moved directly to the more general kernel version.

Proposition 6.20 *Suppose that we wish to perform regression on a training set*

$$S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell)),$$

using the feature space implicitly defined by the kernel $K(\mathbf{x}, \mathbf{z})$, and suppose the parameters $\boldsymbol{\alpha}^*$ solve the following quadratic optimisation problem:

$$\begin{array}{ll} \textit{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} y_i \alpha_i - \varepsilon \sum_{i=1}^{\ell} |\alpha_i| \\ & \quad -\frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j \left(K(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{C} \delta_{ij} \right), \\ \textit{subject to} & \sum_{i=1}^{\ell} \alpha_i = 0. \end{array}$$

Let $f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$, where b^* is chosen so that $f(\mathbf{x}_i) - y_i = -\varepsilon - \alpha_i^*/C$ for any i with $\alpha_i^* > 0$. Then the function $f(\mathbf{x})$ is equivalent to the hyperplane in the feature space implicitly defined by the kernel $K(\mathbf{x}, \mathbf{z})$ that solves the optimisation problem (6.7).

Linear ε-Insensitive Loss

In Theorem 4.30 we must minimise the sum of the linear ε-insensitive losses

$$\frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{\ell} L^{\varepsilon}(\mathbf{x}_i, y_i, f),$$

for some value of the parameter C, which as in the classification case can be seen to control the size of $\|\mathbf{w}\|$ for a fixed training set. The equivalent primal optimisation problem is as follows.

$$\begin{array}{ll} \text{minimise} & \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^{\ell} (\xi_i + \hat{\xi}_i), \\ \text{subject to} & ((\mathbf{w} \cdot \mathbf{x}_i) + b) - y_i \leq \varepsilon + \xi_i, \\ & y_i - ((\mathbf{w} \cdot \mathbf{x}_i) + b) \leq \varepsilon + \hat{\xi}_i, \\ & \xi_i, \hat{\xi}_i \geq 0, \; i = 1, 2, \dots, \ell. \end{array} \tag{6.8}$$

The corresponding dual problem can be derived using the now standard techniques:

$$\begin{array}{ll} \text{maximise} & \sum_{i=1}^{\ell} (\hat{\alpha}_i - \alpha_i) y_i - \varepsilon \sum_{i=1}^{\ell} (\hat{\alpha}_i + \alpha_i) \\ & \quad -\frac{1}{2} \sum_{i,j=1}^{\ell} (\hat{\alpha}_i - \alpha_i)(\hat{\alpha}_j - \alpha_j) \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle, \\ \text{subject to} & 0 \leq \alpha_i, \hat{\alpha}_i \leq C, \; i = 1, \dots, \ell, \\ & \sum_{i=1}^{\ell} (\hat{\alpha}_i - \alpha_i) = 0, \; i = 1, \dots, \ell. \end{array}$$

The corresponding Karush–Kuhn–Tucker complementarity conditions are

$$\begin{array}{ll} \alpha_i \left(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b - y_i - \varepsilon - \xi_i \right) = 0, & i = 1, \dots, \ell, \\ \hat{\alpha}_i \left(y_i - \langle \mathbf{w} \cdot \mathbf{x}_i \rangle - b - \varepsilon - \hat{\xi}_i \right) = 0, & i = 1, \dots, \ell, \\ \xi_i \hat{\xi}_i = 0, \; \alpha_i \hat{\alpha}_i = 0, & i = 1, \dots, \ell, \\ (\alpha_i - C) \xi_i = 0, \; (\hat{\alpha}_i - C) \hat{\xi}_i = 0, & i = 1, \dots, \ell. \end{array}$$

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

Again as mentioned in Remark 6.18 substituting α_i for $\hat{\alpha}_i - \alpha_i$, and taking into account that $\alpha_i \hat{\alpha}_i = 0$, we obtain the following proposition.

Proposition 6.21 *Suppose that we wish to perform regression on a training sample*

$$S = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_\ell, y_\ell)),$$

using the feature space implicitly defined by the kernel $K(\mathbf{x}, \mathbf{z})$, and suppose the parameters $\boldsymbol{\alpha}^*$ solve the following quadratic optimisation problem:

$$\begin{array}{ll} \textit{maximise} & W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} y_i \alpha_i - \varepsilon \sum_{i=1}^{\ell} |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j), \\ \textit{subject to} & \sum_{i=1}^{\ell} \alpha_i = 0, \ -C \le \alpha_i \le C, \ i = 1, \dots, \ell. \end{array}$$

Let $f(\mathbf{x}) = \sum_{i=1}^{\ell} \alpha_i^* K(\mathbf{x}_i, \mathbf{x}) + b^*$, where b^* is chosen so that $f(\mathbf{x}_i) - y_i = -\varepsilon$ for any i with $0 < \alpha_i^* < C$. Then the function $f(\mathbf{x})$ is equivalent to the hyperplane in the feature space implicitly defined by the kernel $K(\mathbf{x}, \mathbf{z})$ that solves the optimisation problem (6.8).

Remark 6.22 If we consider the band of $\pm\varepsilon$ around the function output by the learning algorithm, the points that are not strictly inside the tube are support vectors. Those not touching the tube will have the absolute value of that parameter equal to C.

Remark 6.23 We have again described the most standard optimisations considered. A number of variations have been considered in the literature including considering different norms as well as adapting the optimisation to control the number of points lying outside the ε-band. In this case the number of points is given as an input to the problem rather than the value of ε. References to this and other developments in the use of SVMs for regression will be given at the end of the chapter in Section 6.5.

6.2.2 Kernel Ridge Regression

As mention in Remark 6.19 the case $\varepsilon = 0$ for the quadratic loss corresponds to least squares regression with a weight decay factor. This approach to regression is also known as ridge regression, and we will see shortly that it is equivalent to the techniques derived from Gaussian processes. So we will give it an independent derivation, which highlights the connections with those systems. These systems also ignore the bias term. The (primal) problem can therefore be stated as follows:

$$\begin{array}{ll} \text{minimise} & \lambda \|\mathbf{w}\|^2 + \sum_{i=1}^{\ell} \xi_i^2, \\ \text{subject to} & y_i - \langle \mathbf{w} \cdot \mathbf{x}_i \rangle = \xi_i, \ i = 1, \dots, \ell, \end{array} \tag{6.9}$$

from which we derive the following Lagrangian

$$\text{minimise} \quad L(\mathbf{w}, \xi, \boldsymbol{\alpha}) = \lambda \|\mathbf{w}\|^2 + \sum_{i=1}^{\ell} \xi_i^2 + \sum_{i=1}^{\ell} \alpha_i (y_i - \langle \mathbf{w} \cdot \mathbf{x}_i \rangle - \xi_i).$$

Differentiating and imposing stationarity, we obtain that

$$\mathbf{w} = \frac{1}{2\lambda}\sum_{i=1}^{\ell}\alpha_i\mathbf{x}_i \text{ and } \xi_i = \frac{\alpha_i}{2}.$$

Resubstituting these relations gives the following dual problem:

$$\text{maximise} \quad W(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} y_i\alpha_i - \tfrac{1}{4\lambda}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j\langle\mathbf{x}_i\cdot\mathbf{x}_j\rangle - \tfrac{1}{4}\sum\alpha_i^2,$$

that for convenience we rewrite in vector form:

$$W(\boldsymbol{\alpha}) = \mathbf{y}'\boldsymbol{\alpha} - \frac{1}{4\lambda}\boldsymbol{\alpha}'\mathbf{K}\boldsymbol{\alpha} - \frac{1}{4}\boldsymbol{\alpha}'\boldsymbol{\alpha},$$

where $\mathbf{K}$ denotes the Gram matrix $\mathbf{K}_{ij} = \langle\mathbf{x}_i\cdot\mathbf{x}_j\rangle$, or the kernel matrix $\mathbf{K}_{ij} = K(\mathbf{x}_i,\mathbf{x}_j)$, if we are working in a kernel-induced feature space. Differentiating with respect to $\boldsymbol{\alpha}$ and imposing stationarity we obtain the condition

$$-\frac{1}{2\lambda}\mathbf{K}\boldsymbol{\alpha} - \frac{1}{2}\boldsymbol{\alpha} + \mathbf{y} = 0,$$

giving the solution

$$\boldsymbol{\alpha} = 2\lambda(\mathbf{K}+\lambda\mathbf{I})^{-1}\mathbf{y}$$

and the corresponding regression function

$$f(\mathbf{x}) = \langle\mathbf{w}\cdot\mathbf{x}\rangle = \mathbf{y}'(\mathbf{K}+\lambda\mathbf{I})^{-1}\mathbf{k}$$

where $\mathbf{k}$ is the vector with entries $k_i = \langle\mathbf{x}_i\cdot\mathbf{x}\rangle$, $i = 1,\ldots,\ell$. Hence, we have the following proposition.

Proposition 6.24 *Suppose that we wish to perform regression on a training sample*

$$S = ((\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_\ell,y_\ell)),$$

using the feature space implicitly defined by the kernel $K(\mathbf{x},\mathbf{z})$, and let $f(\mathbf{x}) = \mathbf{y}'(\mathbf{K}+\lambda\mathbf{I})^{-1}\mathbf{k}$, where $\mathbf{K}$ is the $\ell\times\ell$ matrix with entries $\mathbf{K}_{ij} = K(\mathbf{x}_i,\mathbf{x}_j)$ and $\mathbf{k}$ is the vector with entries $\mathbf{k}_i = K(\mathbf{x}_i,\mathbf{x})$. Then the function $f(\mathbf{x})$ is equivalent to the hyperplane in the feature space implicitly defined by the kernel $K(\mathbf{x},\mathbf{z})$ that solves the ridge regression optimisation problem (6.9).

This algorithm has appeared independently under a number of different names. It is also known as Krieging and the solutions are known as regularisation networks, where the regulariser has been implicitly selected by the choice of kernel. We will see in the next subsection that the same function results when we solve the Bayesian learning problem using Gaussian processes.

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

6.2.3 Gaussian Processes

This subsection will bring together the discussion of Bayesian learning from Section 4.6 with the idea of a Gaussian process introduced in Section 3.5. The posterior distribution is given by

$$P(t, \mathbf{t}|\mathbf{x}, S) \propto P(\mathbf{y}|\mathbf{t})P(t, \mathbf{t}|\mathbf{x}, \mathbf{X}),$$

$\mathbf{y}$ are the output values from the training set, which are assumed to be corrupted by noise and $\mathbf{t}$ are the true target output values related to $\mathbf{y}$ by the distribution,

$$P(\mathbf{y}|\mathbf{t}) \propto \exp\left[-\frac{1}{2}(\mathbf{y}-\mathbf{t})'\Omega^{-1}(\mathbf{y}-\mathbf{t})\right],$$

where $\Omega = \sigma^2\mathbf{I}$. The Gaussian process distribution was introduced in Section 3.5 and is defined as

$$\begin{aligned} P(t, \mathbf{t}|\mathbf{x}, \mathbf{X}) &= P_{f\sim\mathscr{D}}\left[(f(\mathbf{x}), f(\mathbf{x}_1), \dots, f(\mathbf{x}_\ell)) = (t, t_1, \dots, t_\ell)\right] \\ &\propto \exp\left(-\frac{1}{2}\hat{\mathbf{t}}'\hat{\Sigma}^{-1}\hat{\mathbf{t}}\right), \end{aligned}$$

where $\hat{\mathbf{t}} = (t, t_1, \dots, t_\ell)'$ and $\hat{\Sigma}$ is indexed with rows and columns from 0 to ℓ. The principal submatrix on the rows 1 to ℓ, is the matrix Σ, where $\Sigma_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ for the covariance function $K(\mathbf{x}, \mathbf{z})$, while the entry $\hat{\Sigma}_{00} = K(\mathbf{x}, \mathbf{x})$ and the entries in the 0 row and column are

$$\hat{\Sigma}_{0i} = \hat{\Sigma}_{i0} = K(\mathbf{x}, \mathbf{x}_i).$$

The distribution of the variable t is the predictive distribution. It is a Gaussian distribution with mean $f(\mathbf{x})$ and variance $V(\mathbf{x})$, given by

$$\begin{aligned} f(\mathbf{x}) &= \mathbf{y}'(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}, \\ V(\mathbf{x}) &= K(\mathbf{x}, \mathbf{x}) - \mathbf{k}'(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{k}, \end{aligned} \tag{6.10}$$

where $\mathbf{K}$ is the $\ell \times \ell$ matrix with entries $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ and $\mathbf{k}$ is the vector with entries $\mathbf{k}_i = K(\mathbf{x}_i, \mathbf{x})$. Hence, the prediction made by the Gaussian process estimate coincides exactly with the ridge regression function of Proposition 6.24, where the parameter λ has been chosen equal to the variance of the noise distribution. This reinforces the relationship between margin slack optimisation and noise in the data. It suggests that optimising the 2-norm (see problem (6.7)) corresponds to an assumption of Gaussian noise, with variance equal to $\frac{1}{C}$.

The Gaussian process also delivers an estimate for the reliability of the prediction in the form of the variance of the predictive distribution. More importantly the analysis can be used to estimate the evidence in favour of a particular choice of covariance function. This can be used to adaptively choose a parameterised kernel function which maximises the evidence and hence is most likely given the data. The covariance or kernel function can be seen as a model of the data and so this provides a principled method for model selection.

6.3 Discussion

This chapter contains the core material of the book. It shows how the learning theory results of Chapter 4 can be used to avoid the difficulties of using linear functions in the high dimensional kernel-induced feature spaces of Chapter 3. We have shown how the optimisation problems resulting from such an approach can be transformed into dual convex quadratic programmes for each of the approaches adopted for both classification and regression. In the regression case the loss function used only penalises errors greater than a threshold ε. Such a loss function typically leads to a sparse representation of the decision rule giving significant algorithmic and representational advantages. If, however, we set $\varepsilon = 0$ in the case of optimising the 2-norm of the margin slack vector, we recover the regressor output by a Gaussian process with corresponding covariance function, or equivalently the ridge regression function. These approaches have the disadvantage that since $\varepsilon = 0$, the sparseness of the representation has been lost.

The type of criterion that is optimised in all of the algorithms we have considered also arises in many other contexts, which all lead to a solution with a dual representation. We can express these criteria in the general form

$$\|f\|_{\mathscr{H}}^2 + C\frac{1}{\ell}\sum_{i=1}^{\ell} L(y_i, f(\mathbf{x}_i)),$$

where L is a loss function, $\|\cdot\|_{\mathscr{H}}$ a regulariser and C is the regularisation parameter. If L is the square loss, this gives rise to regularisation networks of which Gaussian processes are a special case. For this type of problem the solution can always be expressed in the dual form.

In the next chapter we will describe how these optimisation problems can be solved efficiently, frequently making use of the sparseness of the solution when deriving algorithms for very large datasets.

6.4 Exercises

1. What is the relation between the four expressions $W(\boldsymbol{\alpha})$, $\sum_{i=1}^{\ell} \alpha_i$,

$$\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j),$$

and $\frac{1}{\gamma^2}$, when $\boldsymbol{\alpha}$ is the solution of the optimisation problem (6.2) and γ is the corresponding geometric margin? What happens to the optimisation problem if the data are not linearly separable? How is this problem avoided for the soft margin optimisation problems?

2. Derive the dual optimisation problem of the regression problem (6.7), hence demonstrating Proposition 6.20.

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

3. Consider the optimisation problem

$$\begin{array}{ll} \text{minimise}_{\mathbf{w},b} & \langle \mathbf{w} \cdot \mathbf{w} \rangle + C_1 \sum_{i=1}^{\ell} \xi_i + C_2 \sum_{i=1}^{\ell} \xi_i^2, \\ \text{subject to} & y_i \left(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \right) \geq 1 - \xi_i, \, i = 1, \dots, \ell, \\ & \xi_i \geq 0, \, i = 1, \dots, \ell. \end{array}$$

Discuss the effect of varying the parameters C_1 and C_2. Derive the dual optimisation problem.

6.5 Further Reading and Advanced Topics

Support Vector Machines are a very specific class of algorithms, characterised by the use of kernels, the absence of local minima, the sparseness of the solution and the capacity control obtained by acting on the margin, or on other 'dimension independent' quantities such as the number of support vectors. They were invented by Boser, Guyon and Vapnik [19], and first introduced at the Computational Learning Theory (COLT) 1992 conference with the paper [19]. All of these features, however, were already present and had been used in machine learning since the 1960s: large margin hyperplanes in the *input* space were discussed for example by Duda and Hart [35], Cover [28], Vapnik et al. [166] [161], and several statistical mechanics papers (for example [4]); the use of kernels was proposed by Aronszajn [7], Wahba [171], Poggio [116], and others, but it was the paper by Aizermann et al. [1] in 1964 that introduced the geometrical interpretation of the kernels as inner products in a feature space. Similar optimisation techniques were used in pattern recognition by Mangasarian [84], and the sparseness had also already been discussed [28]. See also [57] for related early work. The use of slack variables to overcome the problem of noise and non-separability was also introduced in the 1960s by Smith [143] and improved by Bennett and Mangasarian [15]. However, it was not until 1992 that all of these features were put together to form the maximal margin classifier, the basic Support Vector Machine, and not until 1995 that the soft margin version [27] was introduced: it is surprising how naturally and elegantly all the pieces fit together and complement each other. The papers [138], [10] gave the first rigorous statistical bound on the generalisation of hard margin SVMs, while the paper [141] gives similar bounds for the soft margin algorithms and for the regression case.

After their introduction, an increasing number of researchers have worked on both the algorithmic and theoretical analysis of these systems, creating in just a few years what is effectively a new research direction in its own right, merging concepts from disciplines as distant as statistics, functional analysis, optimisation, as well as machine learning. The soft margin classifier was introduced a few years later by Cortes and Vapnik [27], and in 1995 the algorithm was extended to the regression case [158].

The two recent books written by Vapnik [158, 159] provide a very extensive theoretical background of the field and develop the concept of a Support Vector Machine.

Since most recent advances in kernels, learning theory, and implementation are discussed at the ends of the relevant chapters, we only give a brief overview of some of the improvements recently obtained from the point of view of the overall algorithm. Work has been done on generalisations of the method [88], and extensions to the multi-class case [178], [159], [113]. More general regression scenarios have also been studied: Smola, Schölkopf and Mueller discussed a very general class of loss functions [147], and the use of ridge regression in feature space was considered by [144] and [125].

Ridge regression is a special case of regularisation networks. The concept of regularisation was introduced by Tikhonov [153], and applied to learning in the form of regularisation networks by Girosi et al. [52]. The relation between regularisation networks and Support Vector Machines has been explored by a number of authors [51], [171], [172], [146], [38]. The connection between regularisation networks and neural networks was explored as early as 1990 in [116], see also [52]and [39] for a full bibliography.

The ν-Support Vector algorithm for classification and regression described in Remark 6.13 was introduced in [135] and further developed in [131] and [130]. Other adaptations of the basic approach have been used for density estimation [167], transduction [13], Bayes point estimation [59], ordinal regression [60], etc. Rifkin et al. [121] show that for some degenerate training sets the soft margin gives a trivial solution.

Theoretical advances that do not fit within the framework of Chapter 4 include the analyses of generalisation given in the article [34], which provides a statistical mechanical analysis of SVMs, the papers [66], [160], [177], [173], [107], which provide a cross-validation analysis of the expected error, and the book [159], which gives an expected error bound in terms of the margin and radius of the smallest ball containing the essential support vectors.

Extensions of the SVM concept have been made by several authors, for example Mangasarian's generalised SVMs [83]. Particularly interesting is the development of a system, called the Bayes point machine [59], that enforces another inductive principle, given by Bayesian generalisation theory. Albeit losing the feature of sparseness, this system exhibits excellent performance, and illustrates the important point that this class of algorithms is not limited to the use of margin bounds. Another example of a system similar to SVMs that does not enforce a margin bound is given by Gaussian processes.

More recently, a number of practical applications of SVMs have been reported, in fields as diverse as bioinformatics, computational linguistics and computer vision (some of which are reported in Chapter 8). Many of these recent advances are reported in the collections [132], and [149], and in the surveys [23], [145], [39]. Most of the new contributions are only available from the internet, and can be accessed via the website [30].

Finally, the PhD dissertations of Cortes [26], Schölkopf [129] and Smola [148], provide a valuable first hand source of research topics including work on the feasibility gap.

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press

Gaussian processes are surveyed in [180], and in the dissertation of Rasmussen [120]. Extensions of Gaussian processes to the classification case have also been undertaken but fall beyond the scope of this book.

These references are also given on the website **www.support-vector.net**, which will be kept up to date with new work, pointers to software and papers that are available on-line.

https://doi.org/10.1017/CBO9780511801389.008 Published online by Cambridge University Press