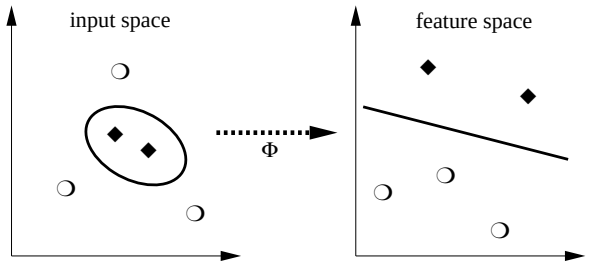

Figure 1.6 The idea of SVMs: map the training data into a higher-dimensional feature space via Φ, and construct a separating hyperplane with maximum margin there. This yields a nonlinear decision boundary in input space. By the use of a kernel function (1.2), it is possible to compute the separating hyperplane without explicitly carrying out the map into the feature space.

1.5 Support Vector Classification

We now have all the tools to describe SVMs (Figure 1.6). Everything in the last section was formulated in a dot product space. We think of this space as the feature space $\mathcal{H}$ of Section 1.1. To express the formulas in terms of the input patterns in $\mathcal{X}$, we thus need to employ (1.6), which expresses the dot product of bold face feature vectors $\mathbf{x}, \mathbf{x}'$ in terms of the kernel k evaluated on input patterns x, x',

$$k(x, x') = \langle \mathbf{x}, \mathbf{x}' \rangle . \tag{1.34}$$

This substitution, which is sometimes referred to as the *kernel trick*, was used by Boser, Guyon, and Vapnik [62] to extend the Generalized Portrait hyperplane classifier to nonlinear Support Vector Machines. Aizerman, Braverman, and Rozonoér [4] called $\mathcal{H}$ the *linearization space*, and used it in the context of the potential function classification method to express the dot product between elements of $\mathcal{H}$ in terms of elements of the input space.

The kernel trick can be applied since all feature vectors only occurred in dot products (see (1.31) and (1.33)). The weight vector (cf. (1.29)) then becomes an expansion in feature space, and therefore will typically no longer correspond to the Φ-image of a single input space vector (cf. Chapter 18). We obtain decision functions of the form (cf. (1.33))

Decision Function

$$f(x) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i \langle \Phi(x), \Phi(x_i) \rangle + b\right) = \operatorname{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i k(x, x_i) + b\right), \tag{1.35}$$

and the following quadratic program (cf. (1.31)):

$$\underset{\boldsymbol{\alpha} \in \mathbb{R}^m}{\text{maximize}} \quad W(\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i, x_j) \tag{1.36}$$

$$\text{subject to} \quad \alpha_i \geq 0 \text{ for all } i = 1, \ldots, m, \quad \text{and} \quad \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{1.37}$$

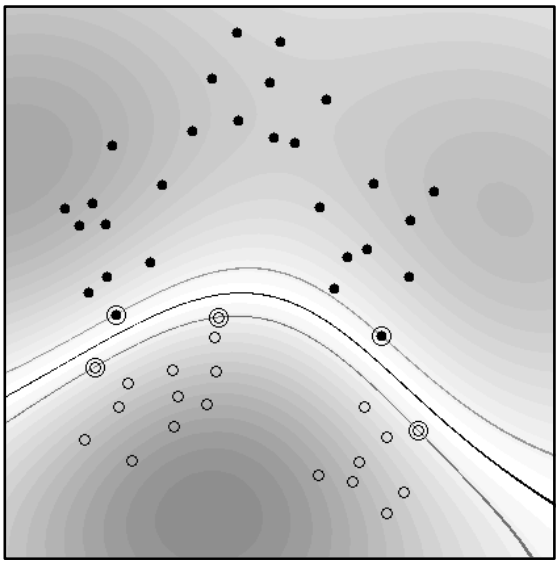

Figure 1.7 Example of an SV classifier found using a radial basis function kernel $k(x, x') = \exp(-\|x - x'\|^2)$ (here, the input space is $\mathcal{X} = [-1, 1]^2$). Circles and disks are two classes of training examples; the middle line is the decision surface; the outer lines precisely meet the constraint (1.25). Note that the SVs found by the algorithm (marked by extra circles) are not centers of clusters, but examples which are critical for the given classification task. Gray values code $|\sum_{i=1}^{m} y_i \alpha_i k(x, x_i) + b|$, the modulus of the argument of the decision function (1.35). The top and the bottom lines indicate places where it takes the value 1 (from [471]).

Figure 1.7 shows an example of this approach, using a Gaussian radial basis function kernel. We will later study the different possibilities for the kernel function in detail (Chapters 2 and 13).

Soft Margin Hyperplane

In practice, a separating hyperplane may not exist, e.g., if a high noise level causes a large overlap of the classes. To allow for the possibility of examples violating (1.25), one introduces slack variables [111, 561, 481]

$$\xi_i \geq 0 \text{ for all } i = 1, \ldots, m, \tag{1.38}$$

in order to relax the constraints (1.25) to

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq 1 - \xi_i \text{ for all } i = 1, \ldots, m. \tag{1.39}$$

A classifier that generalizes well is then found by controlling both the classifier capacity (via $\|\mathbf{w}\|$) and the sum of the slacks $\sum_i \xi_i$. The latter can be shown to provide an upper bound on the number of training errors.

One possible realization of such a *soft margin* classifier is obtained by minimizing the objective function

$$\tau(\mathbf{w}, \boldsymbol{\xi}) = \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_{i=1}^{m} \xi_i \tag{1.40}$$

subject to the constraints (1.38) and (1.39), where the constant $C > 0$ determines the trade-off between margin maximization and training error minimization.[11] Incorporating a kernel, and rewriting it in terms of Lagrange multipliers, this again leads to the problem of maximizing (1.36), subject to the constraints

$$0 \leq \alpha_i \leq C \text{ for all } i = 1, \ldots, m, \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0. \tag{1.41}$$

The only difference from the separable case is the upper bound C on the Lagrange multipliers α_i. This way, the influence of the individual patterns (which could be outliers) gets limited. As above, the solution takes the form (1.35). The threshold b can be computed by exploiting the fact that for all SVs x_i with $\alpha_i < C$, the slack variable ξ_i is zero (this again follows from the KKT conditions), and hence

$$\sum_{j=1}^{m} \alpha_j y_j k(x_i, x_j) + b = y_i. \tag{1.42}$$

Geometrically speaking, choosing b amounts to shifting the hyperplane, and (1.42) states that we have to shift the hyperplane such that the SVs with zero slack variables lie on the ± 1 lines of Figure 1.5.

Another possible realization of a soft margin variant of the optimal hyperplane uses the more natural ν-parametrization. In it, the parameter C is replaced by a parameter $\nu \in (0, 1]$ which can be shown to provide lower and upper bounds for the fraction of examples that will be SVs and those that will have non-zero slack variables, respectively. It uses a primal objective function with the error term $\left(\frac{1}{\nu m} \sum_i \xi_i\right) - \rho$ instead of $C \sum_i \xi_i$ (cf. (1.40)), and separation constraints that involve a margin parameter ρ,

$$y_i(\langle \mathbf{w}, \mathbf{x}_i \rangle + b) \geq \rho - \xi_i \text{ for all } i = 1, \ldots, m, \tag{1.43}$$

which itself is a variable of the optimization problem. The dual can be shown to consist in maximizing the quadratic part of (1.36), subject to $0 \leq \alpha_i \leq 1/(\nu m)$, $\sum_i \alpha_i y_i = 0$ and the additional constraint $\sum_i \alpha_i = 1$. We shall return to these methods in more detail in Section 7.5.

1.6 Support Vector Regression

Let us turn to a problem slightly more general than pattern recognition. Rather than dealing with outputs $y \in \{\pm 1\}$, *regression estimation* is concerned with estimating real-valued functions.

To generalize the SV algorithm to the regression case, an analog of the soft margin is constructed in the space of the target values y (note that we now have

11. It is sometimes convenient to scale the sum in (1.40) by C/m rather than C, as done in Chapter 7 below.