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.

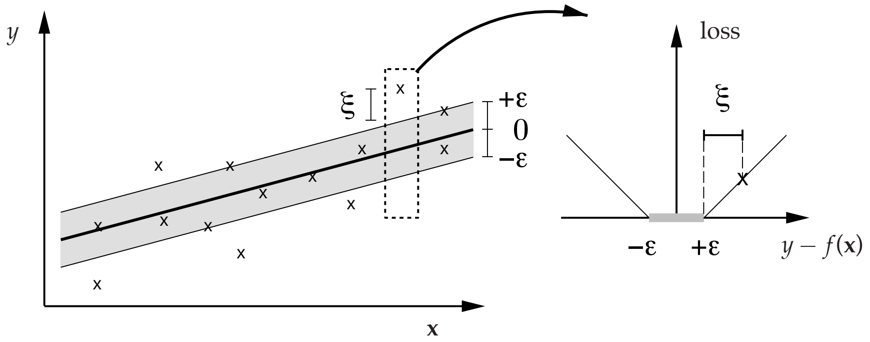

Figure 1.8 In SV regression, a tube with radius ε is fitted to the data. The trade-off between model complexity and points lying outside of the tube (with positive slack variables ξ) is determined by minimizing (1.47).

ε-Insensitive Loss

$y \in \mathbb{R}$) by using Vapnik's *ε-insensitive loss function* [561] (Figure 1.8, see Chapters 3 and 9) . This quantifies the loss incurred by predicting $f(\mathbf{x})$ instead of y as

$$c(x, y, f(x)) := |y - f(\mathbf{x})|_\varepsilon := \max\{0, |y - f(\mathbf{x})| - \varepsilon\}. \tag{1.44}$$

To estimate a linear regression

$$f(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + b, \tag{1.45}$$

one minimizes

$$\frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m} |y_i - f(\mathbf{x}_i)|_\varepsilon. \tag{1.46}$$

Note that the term $\|\mathbf{w}\|^2$ is the same as in pattern recognition (cf. (1.40)); for further details, cf. Chapter 9.

We can transform this into a constrained optimization problem by introducing slack variables, akin to the soft margin case. In the present case, we need two types of slack variable for the two cases $f(\mathbf{x}_i) - y_i > \varepsilon$ and $y_i - f(\mathbf{x}_i) > \varepsilon$. We denote them by $\boldsymbol{\xi}$ and $\boldsymbol{\xi}^*$, respectively, and collectively refer to them as $\boldsymbol{\xi}^{(*)}$.

The optimization problem is given by

$$\underset{\mathbf{w}\in\mathcal{H}, \boldsymbol{\xi}^{(*)}\in\mathbb{R}^m, b\in\mathbb{R}}{\text{minimize}} \quad \tau(\mathbf{w}, \boldsymbol{\xi}^{(*)}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \tag{1.47}$$

$$\text{subject to} \quad f(\mathbf{x}_i) - y_i \le \varepsilon + \xi_i \tag{1.48}$$

$$y_i - f(\mathbf{x}_i) \le \varepsilon + \xi_i^* \tag{1.49}$$

$$\xi_i, \xi_i^* \ge 0 \qquad \text{for all } i = 1, \ldots, m. \tag{1.50}$$

Note that according to (1.48) and (1.49), any error smaller than ε does not require a nonzero ξ_i or ξ_i^* and hence does not enter the objective function (1.47).

Generalization to *kernel*-based regression estimation is carried out in an analogous manner to the case of pattern recognition. Introducing Lagrange multipliers, one arrives at the following optimization problem (for $C, \varepsilon \geq 0$ chosen a priori):

$$\begin{aligned} \underset{\boldsymbol{\alpha},\boldsymbol{\alpha}^* \in \mathbb{R}^m}{\text{maximize}} \quad W(\boldsymbol{\alpha},\boldsymbol{\alpha}^*) &= -\varepsilon \sum_{i=1}^{m} (\alpha_i^* + \alpha_i) + \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) y_i \\ &\quad - \frac{1}{2} \sum_{i,j=1}^{m} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) k(x_i, x_j) \end{aligned} \tag{1.51}$$

$$\text{subject to} \quad 0 \leq \alpha_i, \alpha_i^* \leq C \text{ for all } i = 1, \ldots, m, \text{ and } \sum_{i=1}^{m} (\alpha_i - \alpha_i^*) = 0. \tag{1.52}$$

Regression Function

The regression estimate takes the form

$$f(x) = \sum_{i=1}^{m} (\alpha_i^* - \alpha_i) k(x_i, x) + b, \tag{1.53}$$

where b is computed using the fact that (1.48) becomes an equality with $\xi_i = 0$ if $0 < \alpha_i < C$, and (1.49) becomes an equality with $\xi_i^* = 0$ if $0 < \alpha_i^* < C$ (for details, see Chapter 9). The solution thus looks quite similar to the pattern recognition case (cf. (1.35) and Figure 1.9).

A number of extensions of this algorithm are possible. From an abstract point of view, we just need some target function which depends on $(\mathbf{w}, \boldsymbol{\xi})$ (cf. (1.47)). There are multiple degrees of freedom for constructing it, including some freedom how to penalize, or regularize. For instance, more general loss functions can be used for $\boldsymbol{\xi}$, leading to problems that can still be solved efficiently ([512, 515], cf. Chapter 9). Moreover, norms other than the 2-norm $\|.\|$ can be used to regularize the solution (see Sections 4.9 and 9.4).

ν-SV Regression

Finally, the algorithm can be modified such that ε need not be specified a priori. Instead, one specifies an upper bound $0 \leq \nu \leq 1$ on the fraction of points allowed to lie outside the tube (asymptotically, the number of SVs) and the corresponding ε is computed automatically. This is achieved by using as primal objective function

$$\frac{1}{2}\|\mathbf{w}\|^2 + C \left(\nu m \varepsilon + \sum_{i=1}^{m} |y_i - f(\mathbf{x}_i)|_{\varepsilon} \right) \tag{1.54}$$

instead of (1.46), and treating $\varepsilon \geq 0$ as a parameter over which we minimize. For more detail, cf. Section 9.3.

1.7 Kernel Principal Component Analysis

The kernel method for computing dot products in feature spaces is not restricted to SVMs. Indeed, it has been pointed out that it can be used to develop nonlinear generalizations of any algorithm that can be cast in terms of dot products, such as principal component analysis (PCA) [480].

Principal component analysis is perhaps the most common feature extraction algorithm; for details, see Chapter 14. The term *feature extraction* commonly refers