7.2 The Role of the Margin

The margin plays a crucial role in the design of SV learning algorithms. Let us start by formally defining it.

Definition 7.2 (Geometrical Margin) *For a hyperplane* $\{\mathbf{x} \in \mathcal{H} | \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}$*, we call*

$$\rho_{(\mathbf{w},b)}(\mathbf{x}, y) := y(\langle \mathbf{w}, \mathbf{x} \rangle + b)/\|\mathbf{w}\| \tag{7.4}$$

Geometrical Margin

the geometrical margin of the point $(\mathbf{x}, y) \in \mathcal{H} \times \{\pm 1\}$. *The minimum value*

$$\rho_{(\mathbf{w},b)} := \min_{i=1,\ldots,m} \rho_{(\mathbf{w},b)}(\mathbf{x}_i, y_i) \tag{7.5}$$

shall be called the geometrical margin of $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$*. If the latter is omitted, it is understood that the training set is meant.*

Occasionally, we will omit the qualification *geometrical*, and simply refer to the *margin*.

For a point $(\mathbf{x}, y)$ which is correctly classified, the margin is simply the distance from $\mathbf{x}$ to the hyperplane. To see this, note first that the margin is zero *on* the hyperplane. Second, in the definition, we effectively consider a hyperplane

$$(\hat{\mathbf{w}}, \hat{b}) := (\mathbf{w}/\|\mathbf{w}\|, b/\|\mathbf{w}\|), \tag{7.6}$$

which has a unit length weight vector, and then compute the quantity $y(\langle \hat{\mathbf{w}}, \mathbf{x} \rangle + \hat{b})$. The term $\langle \hat{\mathbf{w}}, \mathbf{x} \rangle$, however, simply computes the length of the projection of $\mathbf{x}$ onto the direction orthogonal to the hyperplane, which, after adding the offset $\hat{b}$, equals the distance to it. The multiplication by y ensures that the margin is positive whenever a point is correctly classified. For misclassified points, we thus get a margin which equals the *negative* distance to the hyperplane. Finally, note that for canonical hyperplanes, the margin is $1/\|\mathbf{w}\|$ (Figure 7.2). The definition of the canonical hyperplane thus ensures that the length of $\mathbf{w}$ now corresponds to a meaningful geometrical quantity.

Margin of Canonical Hyperplanes

It turns out that the margin of a separating hyperplane, and thus the length of the weight vector $\mathbf{w}$, plays a fundamental role in support vector type algorithms. Loosely speaking, if we manage to separate the training data with a large margin, then we have reason to believe that we will do well on the test set. Not surprisingly, there exist a number of explanations for this intuition, ranging from the simple to the rather technical. We will now briefly sketch some of them.

Insensitivity to Pattern Noise

The simplest possible justification for large margins is as follows. Since the training and test data are assumed to have been generated by the same underlying dependence, it seems reasonable to assume that most of the test patterns will lie close (in $\mathcal{H}$) to at least one of the training patterns. For the sake of simplicity, let us consider the case where *all* test points are generated by adding bounded pattern noise (sometimes called input noise) to the training patterns. More precisely, given a training point $(\mathbf{x}, y)$, we will generate test points of the form $(\mathbf{x} + \Delta\mathbf{x}, y)$, where

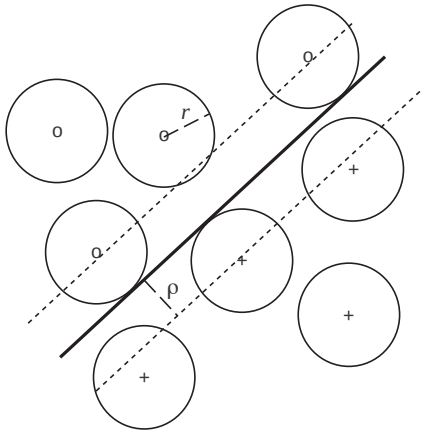

Figure 7.3 Two-dimensional toy example of a classification problem: Separate 'o' from '+' using a hyperplane. Suppose that we add bounded noise to each pattern. If the optimal margin hyperplane has margin ρ, and the noise is bounded by $r < \rho$, then the hyperplane will correctly separate even the noisy patterns. Conversely, if we ran the perceptron algorithm (which finds *some* separating hyperplane, but not necessarily the optimal one) on the noisy data, then we would recover the optimal hyperplane in the limit $r \to \rho$.

$\Delta\mathbf{x} \in \mathcal{H}$ is bounded in norm by some $r > 0$. Clearly, if we manage to separate the training set with a margin $\rho > r$, we will correctly classify *all* test points: Since all training points have a distance of at least ρ to the hyperplane, the test patterns will still be on the correct side (Figure 7.3, cf. also [152]).

If we knew ρ beforehand, then this could actually be turned into an optimal margin classifier training algorithm, as follows. If we use an r which is slightly smaller than ρ, then even the patterns with added noise will be separable with a nonzero margin. In this case, the standard perceptron algorithm can be shown to converge.[1]

Therefore, we can run the perceptron algorithm on the noisy patterns. If the algorithm finds a sufficient number of noisy versions of each pattern, with different perturbations $\Delta\mathbf{x}$, then the resulting hyperplane will not intersect any of the balls depicted in Figure 7.3. As r approaches ρ, the resulting hyperplane should better approximate the maximum margin solution (the figure depicts the limit $r = \rho$). This constitutes a connection between training with pattern noise and maximizing the margin. The latter, in turn, can be thought of as a regularizer, comparable to those discussed earlier (see Chapter 4 and (2.49)). Similar connections to training with noise, for other types of regularizers, have been pointed out before for neural networks [50].

1. Rosenblatt's perceptron algorithm [439] is one of the simplest conceivable iterative procedures for computing a separating hyperplane. In its simplest form, it proceeds as follows. We start with an arbitrary weight vector $\mathbf{w}_0$. At step $n \in \mathbb{N}$, we consider the training example $(\mathbf{x}_n, y_n)$. If it is classified correctly using the current weight vector (i.e., if $\operatorname{sgn}\langle \mathbf{x}_n, \mathbf{w}_{n-1}\rangle = y_n$), we set $\mathbf{w}_n := \mathbf{w}_{n-1}$; otherwise, we set $\mathbf{w}_n := \mathbf{w}_{n-1} + \eta y_i \mathbf{x}_i$ (here, $\eta > 0$ is a learning rate). We thus loop over all patterns repeatedly, until we can complete one full pass through the training set without a single error. The resulting weight vector will thus classify all points correctly. Novikoff [386] proved that this procedure terminates, provided that the training set is separable with a nonzero margin.

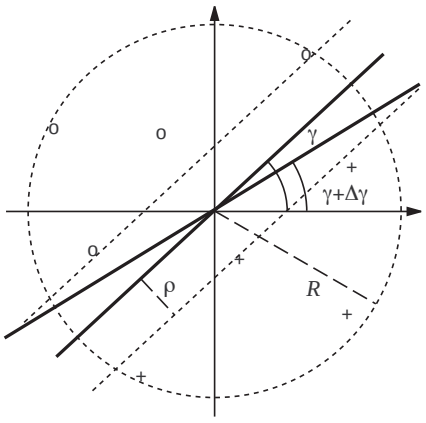

Figure 7.4 Two-dimensional toy example of a classification problem: Separate 'o' from '+' using a hyperplane passing through the origin. Suppose the patterns are bounded in length (distance to the origin) by R, and the classes are separated by an optimal hyperplane (parametrized by the angle γ) with margin ρ. In this case, we can perturb the parameter by some $\Delta\gamma$ with $|\Delta\gamma| < \arcsin \frac{\rho}{R}$, and still correctly separate the data.

Parameter Noise

A similar robustness argument can be made for the dependence of the hyperplane on the parameters $(\mathbf{w}, b)$ (cf. [504]). If all points lie at a distance of at least ρ from the hyperplane, and the patterns are bounded in length, then small perturbations to the hyperplane parameters will not change the classification of the training data (see Figure 7.4).[2] Being able to perturb the parameters of the hyperplane amounts to saying that to store the hyperplane, we need fewer bits than we would for a hyperplane whose *exact* parameter settings are crucial. Interestingly, this is related to what is called the Minimum Description Length principle ([583, 433, 485], cf. also [522, 305, 94]): The best description of the data, in terms of generalization error, should be the one that requires the fewest bits to store.

VC Margin Bound

We now move on to a more technical justification of large margin algorithms. For simplicity, we only deal with hyperplanes that have offset $b = 0$, leaving $f(\mathbf{x}) = \operatorname{sgn} \langle \mathbf{w}, \mathbf{x} \rangle$. The theorem below follows from a result in [24].

Theorem 7.3 (Margin Error Bound) *Consider the set of decision functions* $f(\mathbf{x}) = \operatorname{sgn} \langle \mathbf{w}, \mathbf{x} \rangle$ *with* $\|\mathbf{w}\| \leq \Lambda$ *and* $\|\mathbf{x}\| \leq R$, *for some* $R, \Lambda > 0$. *Moreover, let* $\rho > 0$, *and* ν *denote the fraction of training examples with margin smaller than* $\rho / \|\mathbf{w}\|$, *referred to as the* margin error.

Margin Error

For all distributions P *generating the data, with probability at least* $1 - \delta$ *over the drawing of the* m *training patterns, and for any* $\rho > 0$ *and* $\delta \in (0, 1)$, *the probability that a test pattern drawn from* P *will be misclassified is bounded from above, by*

$$\nu + \sqrt{\frac{c}{m} \left(\frac{R^2 \Lambda^2}{\rho^2} \ln^2 m + \ln(1/\delta) \right)}. \tag{7.7}$$

Here, c *is a universal constant.*

2. Note that this would not hold true if we allowed patterns of arbitrary length — this type of restriction of the pattern lengths pops up in various places, such as Novikoff's theorem [386], Vapnik's VC dimension bound for margin classifiers (Theorem 5.5), and Theorem 7.3.

Let us try to understand this theorem. It makes a probabilistic statement about a probability, by giving an upper bound on the probability of test error, which *itself* only holds true with a certain probability, $1-\delta$. Where do these two probabilities come from? The first is due to the fact that the *test* examples are randomly drawn from P; the second is due to the *training* examples being drawn from P. Strictly speaking, the bound does not refer to a *single* classifier that has been trained on some fixed data set at hand, but to an ensemble of classifiers, trained on various instantiations of training sets generated by the same underlying regularity P.

It is beyond the scope of the present chapter to prove this result. The basic ingredients of bounds of this type, commonly referred to as *VC bounds*, are described in Chapter 5; for further details, see Chapter 12, and [562, 491, 504, 125]. Several aspects of the bound are noteworthy. The test error is bounded by a sum of the margin error ν, and a capacity term (the $\sqrt{\ldots}$ term in (7.7)), with the latter tending to zero as the number of examples, m, tends to infinity. The capacity term can be kept small by keeping R and Λ small, and making ρ large. If we assume that R and Λ are fixed a priori, the main influence is ρ. As can be seen from (7.7), a large ρ leads to a small capacity term, but the margin error ν gets larger. A small ρ, on the other hand, will usually cause fewer points to have margins smaller than $\rho/\|\mathbf{w}\|$, leading to a smaller margin error; but the capacity penalty will increase correspondingly. The overall message: Try to find a hyperplane which is aligned such that even for a large ρ, there are few margin errors.

Maximizing ρ, however, is the same as minimizing the length of $\mathbf{w}$. Hence we might just as well keep ρ fixed, say, equal to 1 (which is the case for canonical hyperplanes), and search for a hyperplane which has a small $\|\mathbf{w}\|$ and few points with a margin smaller than $1/\|\mathbf{w}\|$; in other words (Definition 7.2), few points such that $y\,\langle \mathbf{w}, \mathbf{x}\rangle < 1$.

It should be emphasized that dropping the condition $\|\mathbf{w}\| \le \Lambda$ would prevent us from stating a bound of the kind shown above. We could give an alternative bound, where the capacity depends on the dimensionality of the space $\mathcal{H}$. The crucial advantage of the bound given above is that it is independent of that dimensionality, enabling us to work in very high dimensional spaces. This will become important when we make use of the kernel trick.

It has recently been pointed out that the margin also plays a crucial role in improving asymptotic rates in nonparametric estimation [551]. This topic, however, is beyond the scope of the present book.

Implementation in Hardware

To conclude this section, we note that large margin classifiers also have advantages of a practical nature: An algorithm that can separate a dataset with a certain margin will behave in a benign way when implemented in hardware. Real-world systems typically work only within certain accuracy bounds, and if the classifier is insensitive to small changes in the inputs, it will usually tolerate those inaccuracies.

We have thus accumulated a fair amount of evidence in favor of the following approach: Keep the margin training error small, and the margin large, in order to achieve high generalization ability. In other words, hyperplane decision functions

should be constructed such that they maximize the margin, and at the same time separate the training data with as few exceptions as possible. Sections 7.3 and 7.5 respectively will deal with these two issues.

7.3 Optimal Margin Hyperplanes

Let us now derive the optimization problem to be solved for computing the optimal hyperplane. Suppose we are given a set of examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$, $\mathbf{x}_i \in \mathcal{H}$, $y_i \in \{\pm 1\}$. Here and below, the index i runs over $1, \ldots, m$ by default. We assume that there is at least one negative and one positive y_i. We want to find a decision function $f_{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x}\rangle + b\right)$ satisfying

$$f_{\mathbf{w},b}(\mathbf{x}_i) = y_i. \tag{7.8}$$

If such a function exists (the non-separable case will be dealt with later), canonicality (7.2) implies

$$y_i \left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) \geq 1. \tag{7.9}$$

As an aside, note that out of the two canonical forms of the same hyperplane, $(\mathbf{w}, b)$ and $(-\mathbf{w}, -b)$, only one will satisfy equations (7.8) and (7.11). The existence of class labels thus allows to distinguish two orientations of a hyperplane.

Following the previous section, a separating hyperplane which generalizes well can thus be constructed by solving the following problem:

$$\underset{\mathbf{w}\in\mathcal{H}, b\in\mathbb{R}}{\text{minimize}} \quad \tau(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2, \tag{7.10}$$

$$\text{subject to} \quad y_i \left(\langle \mathbf{x}_i, \mathbf{w}\rangle + b\right) \geq 1 \text{ for all } i = 1, \ldots, m. \tag{7.11}$$

This is called the *primal optimization problem*.

Problems like this one are the subject of optimization theory. For details on how to solve them, see Chapter 6; for a short intuitive explanation, cf. the remarks following (1.26) in the introductory chapter. We will now derive the so-called *dual problem*, which can be shown to have the same solutions as (7.10). In the present case, it will turn out that it is more convenient to deal with the dual. To derive it, we introduce the Lagrangian,

Lagrangian

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{m} \alpha_i \left(y_i(\langle \mathbf{x}_i, \mathbf{w}\rangle + b) - 1\right), \tag{7.12}$$

with Lagrange multipliers $\alpha_i \geq 0$. Recall that as in Chapter 1, we use bold face Greek variables to refer to the corresponding vectors of variables, for instance, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_m)$.

The Lagrangian L must be maximized with respect to α_i, and minimized with respect to $\mathbf{w}$ and b (see Theorem 6.26). Consequently, at this saddle point, the