7 Pattern Recognition

This chapter is devoted to a detailed description of SV classification (SVC) methods. We have already briefly visited the SVC algorithm in Chapter 1. There will be some overlap with that chapter, but here we give a more thorough treatment.

Overview

We start by describing the classifier that forms the basis for SVC, the separating hyperplane (Section 7.1). Separating hyperplanes can differ in how large a margin of separation they induce between the classes, with corresponding consequences on the generalization error, as discussed in Section 7.2. The "optimal" margin hyperplane is defined in Section 7.3, along with a description of how to compute it. Using the kernel trick of Chapter 2, we generalize to the case where the optimal margin hyperplane is not computed in input space, but in a feature space nonlinearly related to the latter (Section 7.4). This dramatically increases the applicability of the approach, as does the introduction of slack variables to deal with outliers and noise in the data (Section 7.5). Many practical problems require us to classify the data into more than just two classes. Section 7.6 describes how multi-class SV classification systems can be built. Following this, Section 7.7 describes some variations on standard SV classification algorithms, differing in the regularizers and constraints that are used. We conclude with a fairly detailed section on experiments and applications (Section 7.8).

Prerequisites

This chapter requires basic knowledge of kernels, as conveyed in the first half of Chapter 2. To understand details of the optimization problems, it is helpful (but not indispensable) to get some background from Chapter 6. To understand the connections to learning theory, in particular regarding the statistical basis of the regularizer used in SV classification, it would be useful to have read Chapter 5.

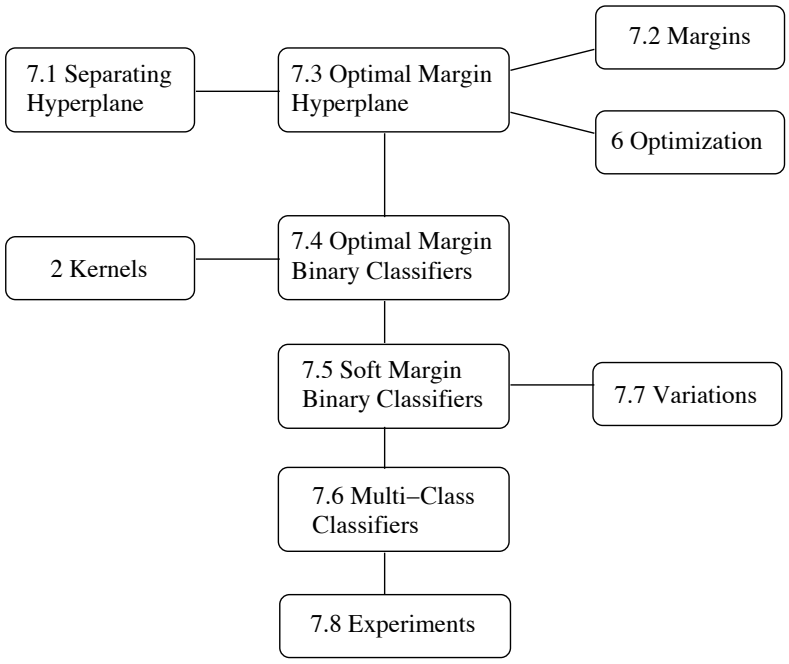
7.1 Separating Hyperplanes

Hyperplane

Suppose we are given a dot product space $\mathcal{H}$, and a set of pattern vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{H}$. Any hyperplane in $\mathcal{H}$ can be written as

$$\{\mathbf{x} \in \mathcal{H} \mid \langle \mathbf{w}, \mathbf{x} \rangle + b = 0\}, \quad \mathbf{w} \in \mathcal{H}, b \in \mathbb{R}. \tag{7.1}$$

In this formulation, $\mathbf{w}$ is a vector orthogonal to the hyperplane: If $\mathbf{w}$ has unit length, then $\langle \mathbf{w}, \mathbf{x} \rangle$ is the length of $\mathbf{x}$ along the direction of $\mathbf{w}$ (Figure 7.1). For general $\mathbf{w}$, this number will be scaled by $\|\mathbf{w}\|$. In any case, the set (7.1) consists



of vectors that all have the same length along $\mathbf{w}$. In other words, these are vectors that project onto the same point on the line spanned by $\mathbf{w}$.

In this formulation, we still have the freedom to multiply $\mathbf{w}$ and b by the same non-zero constant. This superfluous freedom — physicists would call it a "gauge" freedom — can be abolished as follows.

Definition 7.1 (Canonical Hyperplane) *The pair* $(\mathbf{w}, b) \in \mathcal{H} \times \mathbb{R}$ *is called a* canonical *form of the hyperplane (7.1) with respect to* $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathcal{H}$*, if it is scaled such that*

$$\min_{i=1,\ldots,m} |\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1, \tag{7.2}$$

which amounts to saying that the point closest to the hyperplane has a distance of $1/\|\mathbf{w}\|$ *(Figure 7.2).*

Note that the condition (7.2) still allows two such pairs: given a canonical hyperplane $(\mathbf{w}, b)$, another one satisfying (7.2) is given by $(-\mathbf{w}, -b)$. For the purpose of pattern recognition, these two hyperplanes turn out to be different, as they are oriented differently; they correspond to two *decision functions*,

Decision Function

$$\begin{aligned} f_{\mathbf{w},b} : \mathcal{H} &\to \{\pm 1\} \\ \mathbf{x} &\mapsto f_{\mathbf{w},b}(\mathbf{x}) = \operatorname{sgn}\left(\langle \mathbf{w}, \mathbf{x} \rangle + b\right), \end{aligned} \tag{7.3}$$

which are the inverse of each other.

In the absence of class labels $y_i \in \{\pm 1\}$ associated with the $\mathbf{x}_i$, there is no way of distinguishing the two hyperplanes. For a *labelled* dataset, a distinction exists: The two hyperplanes make opposite class assignments. In pattern recognition,

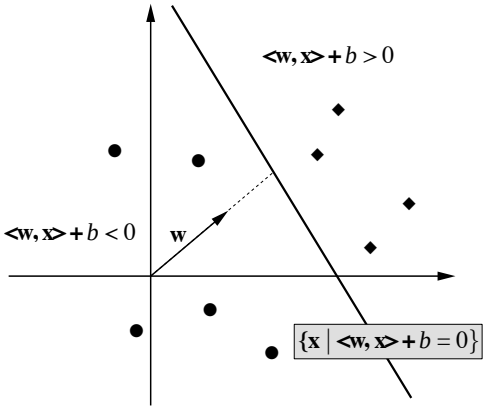


Figure 7.1 A separable classification problem, along with a separating hyperplane, written in terms of an orthogonal weight vector $\mathbf{w}$ and a threshold b. Note that by multiplying both $\mathbf{w}$ and b by the same non-zero constant, we obtain the same hyperplane, represented in terms of different parameters. Figure 7.2 shows how to eliminate this scaling freedom.

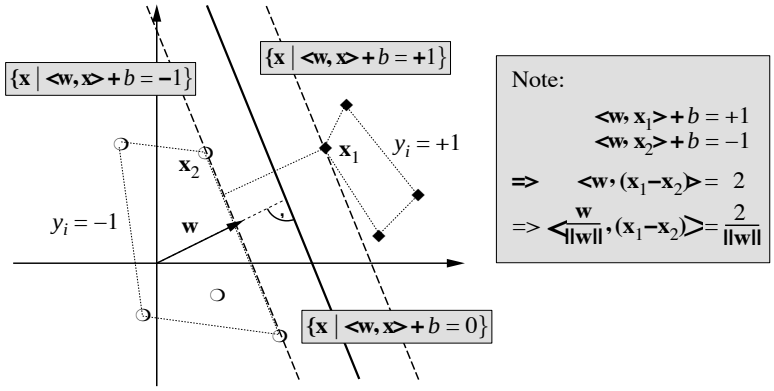


Figure 7.2 By requiring the scaling of $\mathbf{w}$ and b to be such that the point(s) closest to the hyperplane satisfy $|\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1$, we obtain a *canonical* form $(\mathbf{w}, b)$ of a hyperplane. Note that in this case, the margin, measured perpendicularly to the hyperplane, equals $1/\|\mathbf{w}\|$. This can be seen by considering two opposite points which precisely satisfy $|\langle \mathbf{w}, \mathbf{x}_i \rangle + b| = 1$ (cf. Problem 7.4)

we attempt to find a solution $f_{\mathbf{w},b}$ which *correctly classifies* the labelled examples $(\mathbf{x}_i, y_i) \in \mathcal{H} \times \{\pm 1\}$; in other words, which satisfies $f_{\mathbf{w},b}(\mathbf{x}_i) = y_i$ for all i (in this case, the training set is said to be *separable*), or at least for a large fraction thereof.

The next section will introduce the term *margin*, to denote the distance to a separating hyperplane from the point closest to it. It will be argued that to generalize well, a large margin should be sought. In view of Figure 7.2, this can be achieved by keeping $\|\mathbf{w}\|$ small. Readers who are content with this level of detail may skip the next section and proceed directly to Section 7.3, where we describe how to construct the hyperplane with the largest margin.

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