5.11 *Support Vector Machines

We have seen how to train linear machines with margins. *Support Vector Machines* (SVMs) are motivated by many of the same considerations, but rely on preprocessing the data to represent patterns in a high dimension — typically much higher than the original feature space. With an appropriate nonlinear mapping $\varphi()$ to a sufficiently high dimension, data from two categories can always be separated by a hyperplane (Problem 27). Here we assume each pattern $\mathbf{x}_k$ has been transformed to $\mathbf{y}_k = \varphi(\mathbf{x}_k)$; we return to the choice of $\varphi()$ below. For each of the n patterns, $k = 1, 2, ..., n$, we let $z_k = \pm 1$, according to whether pattern k is in ω_1 or ω_2. A linear discriminant in an augmented $\mathbf{y}$ space is

$$g(\mathbf{y}) = \mathbf{a}^t \mathbf{y}, \tag{104}$$

where both the weight vector and the transformed pattern vector are augmented (by $a_0 = w_0$ and $y_0 = 1$, respectively). Thus a separating hyperplane insures

$$z_k g(\mathbf{y}_k) \geq 1 \qquad k = 1, ..., n, \tag{105}$$

much as was shown in Fig. 5.8.

In Sect. **??**, the margin was any positive distance from the decision hyperplane. The goal in training a Support Vector Machine is to find the separating hyperplane with the largest margin; we expect that the larger the margin, the better generalization of the classifier. As illustrated in Fig. 5.2 the distance from any hyperplane to a (transformed) pattern $\mathbf{y}$ is $|g(\mathbf{y})|/||\mathbf{a}||$, and assuming that a positive margin b exists, Eq. 105 implies

$$\frac{z_k g(\mathbf{y}_k)}{||\mathbf{a}||} \geq b \qquad k = 1, ..., n; \tag{106}$$

the goal is to find the weight vector $\mathbf{a}$ that maximizes b. Of course, the solution vector can be scaled arbitrarily and still preserve the hyperplane, and thus to insure uniqueness we impose the constraint $b \; ||\mathbf{a}|| \; = \; 1$; that is, we demand the solution to Eqs. 104 & 105 also minimize $||\mathbf{a}||^2$.

SUPPORT VECTOR

The *support vectors* are the (transformed) training patterns for which Eq. 105 represents an equality — that is, the support vectors are (equally) close to the hyperplane (Fig. 5.19). The support vectors are the training samples that define the optimal separating hyperplane and are the most difficult patterns to classify. Informally speaking, they are the patterns most informative for the classification task.

If N_s denotes the total number of support vectors, then for n training patterns the expected value of the generalization error rate is bounded, according to

$$\mathcal{E}_n[error\ rate] \leq \frac{\mathcal{E}_n[N_s]}{n}, \tag{107}$$

where the expectation is over all training sets of size n drawn from the (stationary) distributions describing the categories. This bound is independent of the dimensionality of the space of transformed vectors, determined by $\varphi()$. We will return to this equation in Chap. **??**, but for now we can understand this informally by means of the *leave one out bound*. Suppose we have n points in the training set, and train a Support Vector Machine on $n - 1$ of them, and test on the single remaining point. If that remaining point happens to be a support vector for the full n sample case, then there will be an error; otherwise, there will not. Note that if we can find a

LEAVE-ONE-OUT BOUND

transformation $\varphi()$ that well separates the data — so the expected number of support vectors is small — then Eq. 107 shows that the expected error rate will be lower.

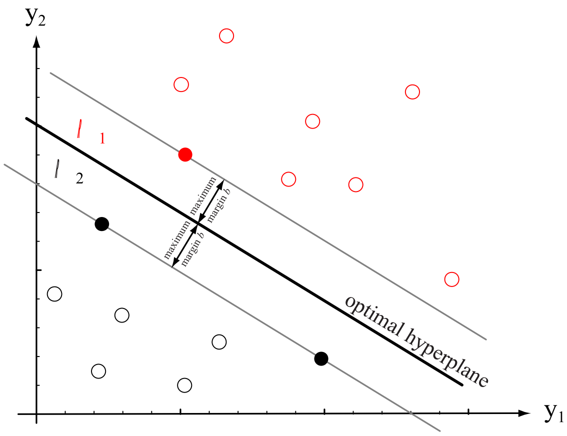


Figure 5.19: Training a Support Vector Machine consists of finding the optimal hyperplane, i.e., the one with the maximum distance from the nearest training patterns. The support vectors are those (nearest) patterns, a distance b from the hyperplane. The three support vectors are shown in solid dots.

5.11.1 SVM training

We now turn to the problem of training an SVM. The first step is, of course, to choose the nonlinear φ-functions that map the input to a higher dimensional space. Often this choice will be informed by the designer's knowledge of the problem domain. In the absense of such information, one might choose to use polynomials, Gaussians or yet other basis functions. The dimensionality of the mapped space can be arbitrarily high (though in practice it may be limited by computational resources).

We begin by recasting the problem of minimizing the magnitude of the weight vector constrained by the separation into an unconstrained problem by the method of Lagrange undetermined multipliers. Thus from Eq. 106 and our goal of minimizing $||\mathbf{a}||$, we construct the functional

$$L(\mathbf{a}, \alpha) = \frac{1}{2}||\mathbf{a}||^2 - \sum_{k=1}^{n} \alpha_k [z_k \mathbf{a}^t \mathbf{y}_k - 1]. \tag{108}$$

and seek to minimize $L()$ with respect to the weight vector $\mathbf{a}$, and maximize it with respect to the undetermined multipliers $\alpha_k \geq 0$. The last term in Eq. 108 expresses the goal of classifying the points correctly. It can be shown using the so-called Kuhn-Tucker construction (Problem 30) (also associated with Karush whose 1939 thesis addressed the same problem) that this optimization can be reformulated as maximizing

$$L(\boldsymbol{\alpha}) = \sum_{k=1}^{n} \alpha_i - \frac{1}{2} \sum_{k,j}^{n} \alpha_k \alpha_j z_k z_j \mathbf{y}_j^t \mathbf{y}_k, \tag{109}$$

subject to the constraints

$$\sum_{k=1}^{n} z_k \alpha_k = 0 \qquad \alpha_k \geq 0, k = 1, ..., n, \tag{110}$$

given the training data. While these equations can be solved using quadratic programming, a number of alternate schemes have been devised (cf. Bibliography).

Example 2: SVM for the XOR problem

The exclusive-OR is the simplest problem that cannot be solved using a linear discriminant operating directly on the features. The points $k = 1, 3$ at $\mathbf{x} = (1, 1)^t$ and $(-1, -1)^t$ are in category ω_1 (red in the figure), while $k = 2, 4$ at $\mathbf{x} = (1, -1)^t$ and $(-1, 1)^t$ are in ω_2 (black in the figure). Following the approach of Support Vector Machines, we preprocess the features to map them to a higher dimension space where they can be linearly separated. While many φ-functions could be used, here we use the simplest expansion up to second order: 1, $\sqrt{2}x_1$, $\sqrt{2}x_2$, $\sqrt{2}x_1x_2$, x_1^2 and x_2^2, where the $\sqrt{2}$ is convenient for normalization.

We seek to maximize Eq. 109,

$$\sum_{k=1}^{4} \alpha_k - \frac{1}{2} \sum_{k,j}^{n} \alpha_k \alpha_j z_k z_j \mathbf{y}_j^t \mathbf{y}_k$$

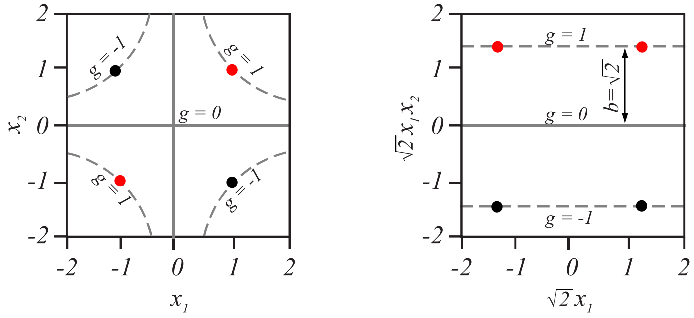
subject to the constraints (Eq. 110)

$$\begin{aligned} \alpha_1 - \alpha_2 + \alpha_3 - \alpha_4 &= 0 \\ 0 \leq \alpha_k & \qquad k = 1, 2, 3, 4. \end{aligned}$$

It is clear from the symmetry of the problem that $\alpha_1 = \alpha_3$ and that $\alpha_2 = \alpha_4$ at the solution. While we could use iterative gradient descent as described in Sect. 5.9, for this small problem we can use analytic techniques instead. The solution is $a_k^* = 1/8$, for $k = 1, 2, 3, 4$, and from the last term in Eq. 108 this implies that all four training patterns are support vectors — an unusual case due to the highly symmetric nature of the XOR problem.

The final discriminant function is $g(\mathbf{x}) = g(x_1, x_2) = x_1 x_2$, and the decision hyperplane is defined by $g = 0$, which properly classifies all training patterns. The margin is easily computed from the solution $||\mathbf{a}||$ and is found to be $b = 1/||\mathbf{a}|| = \sqrt{2}$. The figure at the right shows the margin projected into two dimensions of the five dimensional transformed space. Problem 28 asks you to consider this margin as viewed in other two-dimensional projected sub-spaces.

An important benefit of the Support Vector Machine approach is that the complexity of the resulting classifier is characterized by the number of support vectors — independent of the dimensionality of the transformed space. This



The XOR problem in the original $x_1 - x_2$ feature space is shown at the left; the two red patterns are in category ω_1 and the two black ones in ω_2. These four training patterns $\mathbf{x}$ are mapped to a six-dimensional space by 1, $\sqrt{2}x_1$, $\sqrt{2}x_2$, $\sqrt{2}x_1x_2$, x_1^2 and x_2^2. In this space, the optimal hyperplane is found to be $g(x_1, x_2) = x_1x_2 = 0$ and the margin is $b = \sqrt{2}$. A two-dimensional projection of this space is shown at the right. The hyperplanes through the support vectors are $\sqrt{2}x_1x_2 = \pm 1$, and correspond to the hyperbolas $x_1x_2 = \pm 1$ in the original feature space, as shown.

5.12 Multicategory Generalizations

5.12.1 Kesler's Construction

There is no uniform way to extend all of the two-category procedures we have discussed to the multicategory case. In Sect. 5.2.2 we defined a multicategory classifier called a linear machine which classifies a pattern by computing c linear discriminant functions

$$g_i(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_{i0} \quad i = 1, ..., c,$$

and assigning $\mathbf{x}$ to the category corresponding to the largest discriminant. This is a natural generalization for the multiclass case, particularly in view of the results of Chap. **??** for the multivariate normal problem. It can be extended simply to generalized linear discriminant functions by letting $\mathbf{y}(\mathbf{x})$ be a $\hat{d}$-dimensional vector of functions of $\mathbf{x}$, and by writing

$$g_i(\mathbf{x}) = \mathbf{a}_i^t\mathbf{y} \quad i = 1, ..., c, \tag{111}$$

where again $\mathbf{x}$ is assigned to ω_i if $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all $j \neq i$.

The generalization of our procedures from a two-category linear classifier to a multicategory linear machine is simplest in the linearly-separable case. Suppose that we have a set of labelled samples $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n$, with n_1 in the subset $\mathcal{Y}_1$ labelled ω_1, n_2 in the subset $\mathcal{Y}_2$ labelled ω_2,..., and n_c in the subset $\mathcal{Y}_c$ labelled ω_c. We say that this set is linearly separable if there exists a linear machine that classifies all of them correctly. That is, if these samples are linearly separable, then there exists a set of weight vectors $\hat{\mathbf{a}}_1, ..., \hat{\mathbf{a}}_c$ such that if $\mathbf{y}_k \in \mathcal{Y}_i$, then

$$\hat{\mathbf{a}}_i^t\mathbf{y}_k > \hat{\mathbf{a}}_j^t\mathbf{y}_k \tag{112}$$

for all $j \neq i$.

One of the pleasant things about this definition is that it is possible to manipulate these inequalities and reduce the multicategory problem to the two-category case. Suppose for the moment that $\mathbf{y} \in \mathcal{Y}_1$, so that Eq. 112 becomes

$$\hat{\mathbf{a}}_i^t\mathbf{y}_k - \hat{\mathbf{a}}_j^t\mathbf{y}_k > 0, \quad j = 2, ..., c. \tag{113}$$

This set of $c-1$ inequalities can be thought of as requiring that the $c\hat{d}$-dimensional weight vector

$$\hat{\boldsymbol{\alpha}} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_c \end{bmatrix}$$

correctly classifies all $c-1$ of the $c\hat{d}$-dimensional sample sets

$$\boldsymbol{\eta}_{12} = \begin{bmatrix} \mathbf{y} \\ -\mathbf{y} \\ \mathbf{0} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \quad \boldsymbol{\eta}_{13} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \\ -\mathbf{y} \\ \vdots \\ \mathbf{0} \end{bmatrix}, \quad \cdots, \quad \boldsymbol{\eta}_{1c} = \begin{bmatrix} \mathbf{y} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \\ -\mathbf{y} \end{bmatrix}.$$

In other words, each $\boldsymbol{\eta}_{1j}$ corresponds to "normalizing" the patterns in ω_1 and ω_j. More generally, if $\mathbf{y} \in \mathcal{Y}_i$, we construct $(c-1)c\hat{d}$-dimensional training samples η_{ij} by partitioning η_{ij} into $c\hat{d}$-dimensional subvectors, with the ith subvector being $\mathbf{y}$, the jth being $-\mathbf{y}$, and all others being zero. Clearly, if $\hat{\boldsymbol{\alpha}}^t\boldsymbol{\eta}_{ij} > 0$ for $j \neq i$, then the linear machine corresponding to the components of $\hat{\boldsymbol{\alpha}}$ classifies $\mathbf{y}$ correctly.

This so-called Kesler construction multiplies the dimensionality of the data by c and the number of samples by $c-1$, which does not make its direct use attractive. Its importance resides in the fact that it allows us to convert many multicategory error-correction procedures to two-category procedures for the purpose of obtaining a convergence proof.

5.12.2 Convergence of the Fixed-Increment Rule

We now use use Kesler's construction to prove convergence for a generalization of the fixed-increment rule for a linear machine. Suppose that we have a set of n linearly-separable samples $\mathbf{y}_1, ..., \mathbf{y}_n$, and we use them to form an infinite sequence in which every sample appears infinitely often. Let L_k denote the linear machine whose weight vectors are $\mathbf{a}_1(k), ..., \mathbf{a}_c(k)$. Starting with an arbitrary initial linear machine L_1, we want to use the sequence of samples to construct a sequence of linear machines that converges to a solution machine, one that classifies all of the samples correctly. We shall propose an error-correction rule in which weight changes are made if and only if the present linear machine misclassifies a sample. Let $\mathbf{y}^k$ denote the kth sample requiring correction, and suppose that $\mathbf{y}^k \in \mathcal{Y}_i$. Since $\mathbf{y}^k$ requires correction, there must be at least one $j \neq i$ for which

$$\mathbf{a}_i^t(k)\mathbf{y}^k \leq \mathbf{a}_j(k)^t\mathbf{y}^k. \tag{114}$$

Then the fixed-increment rule for correcting L_k is

$$\left.\begin{array}{rcl}\mathbf{a}_i(k+1) & = & \mathbf{a}_i(k)+\mathbf{y}^k \\ \mathbf{a}_j(k+1) & = & \mathbf{a}_j(k)-\mathbf{y}^k \\ \mathbf{a}_l(k+1) & = & \mathbf{a}_l(k), \qquad l \neq i \quad \text{and } l \neq j.\end{array}\right\} \tag{115}$$

That is, the weight vector for the desired category is incremented by the pattern, the weight vector for the incorrectly chosen category is decremented, and all other weights are left unchanged (Problem 33, Computer exercise 12).

We shall now show that this rule must lead to a solution machine after a finite number of corrections. The proof is simple. For each linear machine L_k there corresponds a weight vector

$$\boldsymbol{\alpha}_k = \begin{bmatrix} \mathbf{a}_1(k) \\ \vdots \\ \mathbf{a}_c(k) \end{bmatrix}.$$

For each sample $\mathbf{y} \in \mathcal{Y}_i$ there are $c-1$ samples η_{ij} formed as described in Sect. **??**. In particular, corresponding to the vector $\mathbf{y}^k$ satisfying Eq. 114 there is a vector

$$\boldsymbol{\eta}_{ij}^k = \begin{bmatrix} \cdot \\ \cdot \\ \cdot \\ \mathbf{y}^k \\ \cdot \\ \cdot \\ \cdot \\ -\mathbf{y}^k \\ \cdot \\ \cdot \\ \cdot \end{bmatrix} \begin{array}{l} \\ \\ \\ \leftarrow i \\ \\ \\ \\ \leftarrow j \\ \\ \\ \\ \end{array}$$

satisfying

$$\boldsymbol{\alpha}^t(k)\boldsymbol{\eta}_{ij}^k \leq 0.$$

Furthermore, the fixed-increment rule for correcting L_k is the fixed-increment rule for correcting $\boldsymbol{\alpha}(k)$, viz.,

$$\boldsymbol{\alpha}(k+1) = \boldsymbol{\alpha}(k) + \boldsymbol{\eta}_{ij}^k.$$

Thus, we have obtained a complete correspondence between the multicategory case and the two-category case, in which the multicategory procedure produces a sequence of samples $\boldsymbol{\eta}^1, \boldsymbol{\eta}^2, ..., \boldsymbol{\eta}^k, ...$ and a sequence of weight vectors $\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_k, ...$ By our results for the the two-cateogry case, this latter sequence can not be infinite, but must terminate in a solution vector. Hence, the sequence $L_1, L_2, ..., L_k, ...$ must terminate in a solution machine after a finite number of corrections.

This use of Kesler's construction to establish equivalences between multicategory and two-category procedures is a powerful theoretical tool. It can be used to extend all of our results for the Perceptron and relaxation procedures to the multicategory case, and applies as well to the error-correction rules for the method of potential functions (Problem **??**). Unfortunately, it is not as directly useful in generalizing the MSE or the linear programming approaches.

5.12.3 Generalizations for MSE Procedures

Perhaps the simplest way to obtain a natural generalization of the MSE procedures to the multiclass case is to consider the problem as a set of c two-class problems. The ith problem is to obtain a weight vector $\mathbf{a}_i$ that is minimum-squared-error solution to the equations

$$\left.\begin{array}{rcl} \mathbf{a}_i^t\mathbf{y} & = & 1 \text{ for all } \mathbf{y} \in \mathcal{Y}_i \\ \mathbf{a}_i^t\mathbf{y} & = & -1 \text{ for all } \mathbf{y} \notin \mathcal{Y}_i. \end{array}\right\}$$

In view of the results of Sect. 5.8.3 the number of samples is very large we will obtain a minimum mean-squared-error approximation to the Bayes discriminant function

$$P(\omega_i|\mathbf{x}) - P(\text{not } \omega_i|\mathbf{x}) = 2P(\omega_i|\mathbf{x}) - 1.$$

This observation has two immediate consequences. First, it suggests a modification in which we seek a weight vector $\mathbf{a}_i$ that is a minimum-squared-error solution to the equations

$$\left.\begin{array}{rcll} \mathbf{a}_i^t\mathbf{y} & = & 1 & \text{for all } \mathbf{y} \in \mathcal{Y}_i \\ \mathbf{a}_i^t\mathbf{y} & = & 0 & \text{for all } \mathbf{y} \notin \mathcal{Y}_i \end{array}\right\} \tag{116}$$

so that $\mathbf{a}^t\mathbf{y}$ will be a minimum mean-squared-error approximation to $P(\omega_i|\mathbf{x})$. Second, it justifies the use of the resulting discriminant functions in a linear machine, in which we assign $\mathbf{y}$ to ω_i if $\mathbf{a}_i^t\mathbf{y} > \mathbf{a}_j^t\mathbf{y}$ for all $j \neq i$.

The pseudoinverse solution to the multiclass MSE problem can be written in a form analogous to the form for the two-class case. Let $\mathbf{Y}$ be the n-by-$\hat{d}$ matrix of training samples, which we assume to be partitioned as

$$\mathbf{Y} = \begin{bmatrix} \mathbf{Y}_1 \\ \mathbf{Y}_2 \\ \vdots \\ \mathbf{Y}_c \end{bmatrix}, \tag{117}$$

with the samples labelled ω_i comprising the rows of $\mathbf{Y}_i$. Similarly, let $\mathbf{A}$ be the $\hat{d}$-by-c matrix of weight vectors

$$\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2\ \cdots\ \mathbf{a}_c], \tag{118}$$

and let $\mathbf{B}$ be the n-by-c matrix

$$\mathbf{B} = \begin{bmatrix} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_c \end{bmatrix}, \tag{119}$$

where all of the elements of $\mathbf{B}_i$ are zero except for those in the ith column, which are unity. Then the trace of the "squared" error matrix $(\mathbf{YA} - \mathbf{B})^t \times (\mathbf{YA} - \mathbf{B})$ is minimized by the solution*

$$\mathbf{A} = \mathbf{Y}^\dagger \mathbf{B}, \tag{120}$$

where, as usual, $\mathbf{Y}^\dagger$ is the pseudoinverse of $\mathbf{Y}$.

This result can be generalized in a theoretically interesting fashion. Let λ_{ij} be the loss incurred for deciding ω_i when the true state of nature is ω_j, and let the jth submatrix of $\mathbf{B}$ be given by

$$\mathbf{B}_j = -\left[\begin{array}{cccc} \lambda_{1j} & \lambda_{2j} & \cdots & \lambda_{cj} \\ \lambda_{1j} & \lambda_{2j} & \cdots & \lambda_{cj} \\ \vdots & & & \vdots \\ \lambda_{1j} & \lambda_{2j} & \cdots & \lambda_{cj} \end{array}\right] \begin{array}{c} \uparrow \\ | \\ n_j \\ | \\ \downarrow \end{array} \qquad j = 1, ..., c. \tag{121}$$

Then, as the number of samples approaches infinity, the solution $\mathbf{A} = \mathbf{Y}^\dagger \mathbf{B}$ yields discriminant functions $\mathbf{a}_i^t \mathbf{y}$ which provide a minimum-mean-square-error approximation to the Bayes discriminant function

$$g_{0i} = -\sum_{j=1}^{c} \lambda_{ij} P(\omega_i|\mathbf{x}). \tag{122}$$

The proof of this is a direct extension of the proof given in Sect. 5.8.3 (Problem 34).

Summary

This chapter considers discriminant functions that are a linear function of a set of parameters, generally called weights. In all two-category cases, such discriminants lead to hyperplane decision boundaries, either in the feature space itself, or in a space where the features have been mapped by a nonlinear function (general linear discriminants).

In broad overview, techniques such as the Perceptron algorithm adjust the parameters to increase the inner product with patterns in category ω_1 and decrease the inner product with those in ω_2. A very general approach is to form some criterion function and perform gradient descent. Different creiterion functions have different strengths and weaknesses related to computation and convergence, none uniformly dominates the others. One can use linear algebra to solve for the weights (parameters) directly, by means of pseudoinverse matrixes for small problems.

In Support Vector Machines, the input is mapped by a nonlinear function to a high-dimensional space, and the optimal hyperplane found, the one that has the largest margin. The support vectors are those (transformed) patterns that determine the margin; they are informally the hardest patterns to classify, and the most informative ones for designing the classifier. An upper bound on expected error rate of the classifier depends linearly upon the expected number of support vectors.

For multi-category problems, the linear machines create decision boundaries consisting of sections of such hyperplanes. One can prove convergence of multi-category

* If we let $\mathbf{b}_i$ denote the ith column of $\mathbf{B}$, the trace of $(\mathbf{YA} - \mathbf{B})^t(\mathbf{YA} - \mathbf{B})$ is equal to the sum of the squared lengths of the error vectors $\mathbf{Ya}_i - \mathbf{b}_i$. The solution $\mathbf{A} = \mathbf{Y}^\dagger \mathbf{B}$ not only minimizes this sum, but it also minimizes each term in the sum.

algorithms by first converting them to two-category algorithms and using the two-category proofs. The simplex algorithm finds the optimimun of a linear function subject to (inequality) constraints, and can be used for training linear classifiers.

Linear discriminants, while useful, are not sufficiently general for arbitrary challenging pattern recognition problems (such as those involving multi-modal densities) unless an appropriate nonlinear mapping (φ function) can be found. In this chapter we have not considered any principled approaches to choosing such functions, but turn to that topic in Chap. **??**.

Bibliographical and Historical Remarks

Because linear discriminant functions are so amenable to analysis, far more papers have been written about them than the subject deserves. Historically, all of this work begins with the classic paper by Ronald A. Fisher [4]. The application of linear discriminant function to pattern classification was well described in [7], which posed the problem of optimal (minimum-risk) linear discriminant, and proposed plausible gradient descient procedures to determine a solution from samples. Unfortunately, little can be said about such procedures without knowing the underlying distributions, and even then the situation is analytically complex. The design of multicategory classifiers using two-category procedures stems from [12]. Minsky and Papert's **Perceptrons** [11] was influential in pointing out the weaknesses of linear classifiers — weaknesses that were overcome by the methods we shall study in Chap. **??**. The Winnow algorithms [8] in the error-free case and [9, 6] and subsequent work in the general case have been useful in the computational learning community, as they allow one to derive convergence bounds.

While this work was statistically oriented, many of the pattern recognition papers that appeared in the late 1950s and early 1960s adopted other viewpoints. One viewpoint was that of neural networks, in which individual neurons were modelled as threshold elements, two-category linear machines — work that had its origins in the famous paper by McCulloch and Pitts [10].

As linear machines have been applied to larger and larger data sets in higher and higher dimensions, the computational burden of linear programming [2] has made this approach less popular. Stochastic approximations, e.g, [15],

An early paper on the key ideas in Support Vector Machines is [1]. A more extensive treatment, including complexity control, can be found in [14] — material we shall visit in Chap. **??**. A readable presentation of the method is [3], which provided the inspiration behind our Example 2. The Kuhn-Tucker construction, used in the SVM training method described in the text and explored in Problem 30, is from [5] and used in [13]. The fundamental result is that exactly one of the following three cases holds. 1) The original (primal) conditions have an optimal solution; in that case the dual cases do too, and their objective values are equal, or 2) the primal conditions are infeasible; in that case the dual is either unbounded or itself infeasible, or 3) the primal conditions are unbounded; in that case the dual is infeasible.

Problems

⊕ *Section 5.2*

1. Consider a linear machine with discriminant functions $g_i(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_{i0}$, $i = 1, ..., c$. Show that the decision regions are convex by showing that if $\mathbf{x}_1 \in \mathcal{R}_i$ and $\mathbf{x}_2 \in \mathcal{R}_i$ then $\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2 \in \mathcal{R}_i$ if $0 \leq \lambda \leq 1$.

2. Figure 5.3 illustrates the two most popular methods for designing a c-category classifier from linear boundary segments. Another method is to save the full $\binom{c}{2}$ linear ω_i/ω_j boundaries, and classify any point by taking a *vote* based on all these boundaries. Prove whether the resulting decision regions must be convex. If they need not be convex, construct a non-pathological example yielding at least one non-convex decision region.

3. Consider the hyperplane used for discriminant functions.

(a) Show that the distance from the hyperplane $g(\mathbf{x}) = \mathbf{w}^t\mathbf{x} + w_0 = 0$ to the point $\mathbf{x}_a$ is $|g(\mathbf{x}_a)|/\|\mathbf{w}\|$ by minimizing $\|\mathbf{x} - \mathbf{x}_a\|^2$ subject to the constraint $g(\mathbf{x}) = 0$.

(b) Show that the projection of $\mathbf{x}_a$ onto the hyperplane is given by

$$\mathbf{x}_p = \mathbf{x}_a - \frac{g(\mathbf{x}_a)}{\|\mathbf{w}\|^2}\mathbf{w}.$$

4. Consider the three-category linear machine with discriminant functions $g_i(\mathbf{x}) = \mathbf{w}_i^t\mathbf{x} + w_{i0}$, $i = 1, 2, 3$.

(a) For the special case where $\mathbf{x}$ is two-dimensional and the threshold weights w_{i0} are zero, sketch the weight vectors with their tails at the origin, the three lines joining their heads, and the decision boundaries.

(b) How does this sketch change when a constant vector $\mathbf{c}$ is added to each of the three weight vectors?

5. In the multicategory case, a set of samples is said to be linearly separable if there exists a linear machine that can classify them all correctly. If any samples labelled ω_i can be separated from all others by a single hyperplane, we shall say the samples are *totally linearly separable*. Show that totally linearly separable samples must be linearly separable, but that the converse need not be true. (Hint: For the converse, consider a case in which a linear machine like the one in Problem 4 separates the samples.)

TOTAL LINEAR SEPARABILITY

6. A set of samples is said to be *pairwise linearly separable* if there exist $c(c-1)/2$ hyperplanes H_{ij} such that H_{ij} separates samples labelled ω_i from samples ω_j. Show that a pairwise-linearly-separable set of patterns may not be linearly separable.

PAIRWISE LINEAR SEPARABILITY

7. Let $\{\mathbf{y}_1, ..., \mathbf{y}_n\}$ be a finite set of linearly separable training samples, and let $\mathbf{a}$ be called a solution vector if $\mathbf{a}^t\mathbf{y}_i \geq 0$ for all i. Show that the minimum-length solution vector is unique. (Hint: Consider the effect of averaging two solution vectors.)

8. The *convex hull* of a set of vectors $\mathbf{x}_i, i = 1, \ldots, n$ is the set of all vectors of the form

CONVEX HULL

$$\mathbf{x} = \sum_{i=1}^{n} \alpha_i \mathbf{x}_i,$$

where the coefficients α_i are nonnegative and sum to one. Given two sets of vectors, show that either they are linearly separable or their convex hulls intersect. (Hint: Suppose that both statements are true, and consider the classification of a point in the intersection of the convex hulls.)