CHAPTER 20 Linear Discriminant Analysis

Given labeled data consisting of d-dimensional points $\mathbf{x}_i$ along with their classes y_i, the goal of linear discriminant analysis (LDA) is to find a vector $\mathbf{w}$ that maximizes the separation between the classes after projection onto $\mathbf{w}$. Recall from Chapter 7 that the first principal component is the vector that maximizes the projected variance of the points. The key difference between principal component analysis and LDA is that the former deals with unlabeled data and tries to maximize variance, whereas the latter deals with labeled data and tries to maximize the discrimination between the classes.

20.1 OPTIMAL LINEAR DISCRIMINANT

Let us assume that the dataset $\mathbf{D}$ consists of n labeled points $\{\mathbf{x}_i, y_i\}$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{c_1, c_2, \ldots, c_k\}$. Let $\mathbf{D}_i$ denote the subset of points labeled with class c_i, i.e., $\mathbf{D}_i = \{\mathbf{x}_j | y_j = c_i\}$, and let $|\mathbf{D}_i| = n_i$ denote the number of points with class c_i. We assume that there are only $k = 2$ classes. Thus, the dataset $\mathbf{D}$ can be partitioned into $\mathbf{D}_1$ and $\mathbf{D}_2$.

Let $\mathbf{w}$ be a unit vector, that is, $\mathbf{w}^T\mathbf{w} = 1$. By Eq. (1.7), the projection of any d-dimensional point $\mathbf{x}_i$ onto the vector $\mathbf{w}$ is given as

$$\mathbf{x}_i' = \left(\frac{\mathbf{w}^T\mathbf{x}_i}{\mathbf{w}^T\mathbf{w}}\right)\mathbf{w} = \left(\mathbf{w}^T\mathbf{x}_i\right)\mathbf{w} = a_i\mathbf{w}$$

where a_i specifies the offset or coordinate of $\mathbf{x}_i'$ along the line $\mathbf{w}$:

$$a_i = \mathbf{w}^T\mathbf{x}_i$$

Thus, the set of n scalars $\{a_1, a_2, \ldots, a_n\}$ represents the mapping from $\mathbb{R}^d$ to $\mathbb{R}$, that is, from the original d-dimensional space to a 1-dimensional space (along $\mathbf{w}$).

Example 20.1. Consider Figure 20.1, which shows the 2-dimensional Iris dataset with `sepal length` and `sepal width` as the attributes, and `iris-setosa` as class c_1 (circles), and the other two Iris types as class c_2 (triangles). There are $n_1 = 50$ points in c_1 and $n_2 = 100$ points in c_2. One possible vector $\mathbf{w}$ is shown, along with the projection

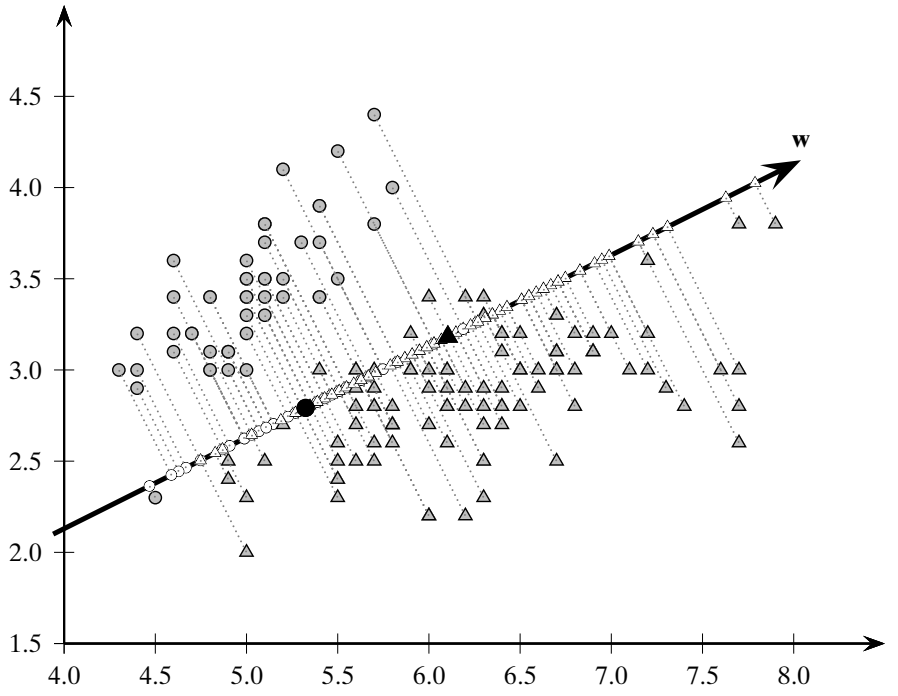

Figure 20.1. Projection onto **w**.

of all the points onto **w**. The projected means of the two classes are shown in black. Here **w** has been translated so that it passes through the mean of the entire data. One can observe that **w** is not very good in discriminating between the two classes because the projection of the points onto **w** are all mixed up in terms of their class labels. The optimal linear discriminant direction is shown in Figure 20.2.

Each point coordinate a_i has associated with it the original class label y_i, and thus we can compute, for each of the two classes, the mean of the projected points as follows:

$$\begin{aligned} m_1 &= \frac{1}{n_1} \sum_{\mathbf{x}_i \in \mathbf{D}_1} a_i \\ &= \frac{1}{n_1} \sum_{\mathbf{x}_i \in \mathbf{D}_1} \mathbf{w}^T \mathbf{x}_i \\ &= \mathbf{w}^T \left(\frac{1}{n_1} \sum_{\mathbf{x}_i \in \mathbf{D}_1} \mathbf{x}_i \right) \\ &= \mathbf{w}^T \boldsymbol{\mu}_1 \end{aligned}$$

where $\boldsymbol{\mu}_1$ is the mean of all point in $\mathbf{D}_1$. Likewise, we can obtain

$$m_2 = \mathbf{w}^T \boldsymbol{\mu}_2$$

In other words, the mean of the projected points is the same as the projection of the mean.

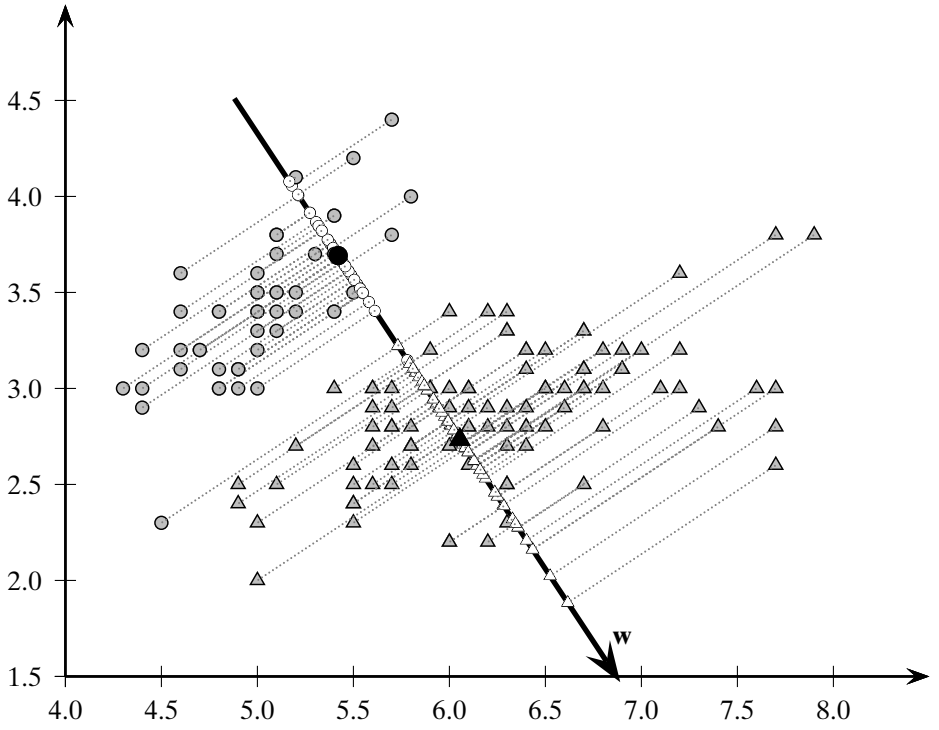

Figure 20.2. Linear discriminant direction **w**.

To maximize the separation between the classes, it seems reasonable to maximize the difference between the projected means, $|m_1 - m_2|$. However, this is not enough. For good separation, the variance of the projected points for each class should also not be too large. A large variance would lead to possible overlaps among the points of the two classes due to the large spread of the points, and thus we may fail to have a good separation. LDA maximizes the separation by ensuring that the *scatter* s_i^2 for the projected points within each class is small, where scatter is defined as

$$s_i^2 = \sum_{\mathbf{x}_j \in \mathbf{D}_i} (a_j - m_i)^2$$

Scatter is the total squared deviation from the mean, as opposed to the variance, which is the average deviation from mean. In other words

$$s_i^2 = n_i \sigma_i^2$$

where $n_i = |\mathbf{D}_i|$ is the size, and σ_i^2 is the variance, for class c_i.

We can incorporate the two LDA criteria, namely, maximizing the distance between projected means and minimizing the sum of projected scatter, into a single maximization criterion called the *Fisher LDA objective*:

$$\max_{\mathbf{w}} \; J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} \tag{20.1}$$

The goal of LDA is to find the vector $\mathbf{w}$ that maximizes $J(\mathbf{w})$, that is, the direction that maximizes the separation between the two means m_1 and m_2, and minimizes the total scatter $s_1^2 + s_2^2$ of the two classes. The vector $\mathbf{w}$ is also called the *optimal linear discriminant (LD)*. The optimization objective [Eq. (20.1)] is in the projected space. To solve it, we have to rewrite it in terms of the input data, as described next.

Note that we can rewrite $(m_1 - m_2)^2$ as follows:

$$\begin{aligned}(m_1 - m_2)^2 &= \left(\mathbf{w}^T(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\right)^2 \\ &= \mathbf{w}^T\left((\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\right)\mathbf{w} \\ &= \mathbf{w}^T\mathbf{B}\mathbf{w} \end{aligned} \tag{20.2}$$

where $\mathbf{B} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$ is a $d \times d$ rank-one matrix called the *between-class scatter matrix*.

As for the projected scatter for class c_1, we can compute it as follows:

$$\begin{aligned} s_1^2 &= \sum_{\mathbf{x}_i \in \mathbf{D}_1} (a_i - m_1)^2 \\ &= \sum_{\mathbf{x}_i \in \mathbf{D}_1} (\mathbf{w}^T\mathbf{x}_i - \mathbf{w}^T\boldsymbol{\mu}_1)^2 \\ &= \sum_{\mathbf{x}_i \in \mathbf{D}_1} \left(\mathbf{w}^T(\mathbf{x}_i - \boldsymbol{\mu}_1)\right)^2 \\ &= \mathbf{w}^T\left(\sum_{\mathbf{x}_i \in \mathbf{D}_1} (\mathbf{x}_i - \boldsymbol{\mu}_1)(\mathbf{x}_i - \boldsymbol{\mu}_1)^T\right)\mathbf{w} \\ &= \mathbf{w}^T\mathbf{S}_1\mathbf{w} \end{aligned} \tag{20.3}$$

where $\mathbf{S}_1$ is the *scatter matrix* for $\mathbf{D}_1$. Likewise, we can obtain

$$s_2^2 = \mathbf{w}^T\mathbf{S}_2\mathbf{w} \tag{20.4}$$

Notice again that the scatter matrix is essentially the same as the covariance matrix, but instead of recording the average deviation from the mean, it records the total deviation, that is,

$$\mathbf{S}_i = n_i\boldsymbol{\Sigma}_i \tag{20.5}$$

Combining Eqs. (20.3) and (20.4), the denominator in Eq. (20.1) can be rewritten as

$$s_1^2 + s_2^2 = \mathbf{w}^T\mathbf{S}_1\mathbf{w} + \mathbf{w}^T\mathbf{S}_2\mathbf{w} = \mathbf{w}^T(\mathbf{S}_1 + \mathbf{S}_2)\mathbf{w} = \mathbf{w}^T\mathbf{S}\mathbf{w} \tag{20.6}$$

where $\mathbf{S} = \mathbf{S}_1 + \mathbf{S}_2$ denotes the *within-class scatter matrix* for the pooled data. Because both $\mathbf{S}_1$ and $\mathbf{S}_2$ are $d \times d$ symmetric positive semidefinite matrices, $\mathbf{S}$ has the same properties.

Using Eqs. (20.2) and (20.6), we write the LDA objective function [Eq. (20.1)] as follows:

$$\max_{\mathbf{w}} \; J(\mathbf{w}) = \frac{\mathbf{w}^T\mathbf{B}\mathbf{w}}{\mathbf{w}^T\mathbf{S}\mathbf{w}} \tag{20.7}$$

To solve for the best direction $\mathbf{w}$, we differentiate the objective function with respect to $\mathbf{w}$, and set the result to zero. We do not explicitly have to deal with the constraint that $\mathbf{w}^T\mathbf{w} = 1$ because in Eq. (20.7) the terms related to the magnitude of $\mathbf{w}$ cancel out in the numerator and the denominator.

Recall that if $f(x)$ and $g(x)$ are two functions then we have

$$\frac{d}{dx}\left(\frac{f(x)}{g(x)}\right) = \frac{f'(x)g(x) - g'(x)f(x)}{g(x)^2}$$

where $f'(x)$ denotes the derivative of $f(x)$. Taking the derivative of Eq. (20.7) with respect to the vector $\mathbf{w}$, and setting the result to the zero vector, gives us

$$\frac{d}{d\mathbf{w}}J(\mathbf{w}) = \frac{2\mathbf{B}\mathbf{w}(\mathbf{w}^T\mathbf{S}\mathbf{w}) - 2\mathbf{S}\mathbf{w}(\mathbf{w}^T\mathbf{B}\mathbf{w})}{(\mathbf{w}^T\mathbf{S}\mathbf{w})^2} = \mathbf{0}$$

which yields

$$\begin{aligned}
\mathbf{B}\,\mathbf{w}(\mathbf{w}^T\mathbf{S}\mathbf{w}) &= \mathbf{S}\,\mathbf{w}(\mathbf{w}^T\mathbf{B}\mathbf{w}) \\
\mathbf{B}\,\mathbf{w} &= \mathbf{S}\,\mathbf{w}\left(\frac{\mathbf{w}^T\mathbf{B}\mathbf{w}}{\mathbf{w}^T\mathbf{S}\mathbf{w}}\right) \\
\mathbf{B}\,\mathbf{w} &= J(\mathbf{w})\mathbf{S}\mathbf{w} \\
\mathbf{B}\mathbf{w} &= \lambda\mathbf{S}\mathbf{w} \qquad (20.8)
\end{aligned}$$

where $\lambda = J(\mathbf{w})$. Eq. (20.8) represents a *generalized eigenvalue problem* where λ is a generalized eigenvalue of $\mathbf{B}$ and $\mathbf{S}$; the eigenvalue λ satisfies the equation $\det(\mathbf{B} - \lambda\mathbf{S}) = 0$. Because the goal is to maximize the objective [Eq. (20.7)], $J(\mathbf{w}) = \lambda$ should be chosen to be the largest generalized eigenvalue, and $\mathbf{w}$ to be the corresponding eigenvector. If $\mathbf{S}$ is *nonsingular*, that is, if $\mathbf{S}^{-1}$ exists, then Eq. (20.8) leads to the regular eigenvalue–eigenvector equation, as

$$\begin{aligned}
\mathbf{B}\mathbf{w} &= \lambda\mathbf{S}\mathbf{w} \\
\mathbf{S}^{-1}\mathbf{B}\mathbf{w} &= \lambda\mathbf{S}^{-1}\mathbf{S}\mathbf{w} \\
(\mathbf{S}^{-1}\mathbf{B})\mathbf{w} &= \lambda\mathbf{w} \qquad (20.9)
\end{aligned}$$

Thus, if $\mathbf{S}^{-1}$ exists, then $\lambda = J(\mathbf{w})$ is an eigenvalue, and $\mathbf{w}$ is an eigenvector of the matrix $\mathbf{S}^{-1}\mathbf{B}$. To maximize $J(\mathbf{w})$ we look for the largest eigenvalue λ, and the corresponding dominant eigenvector $\mathbf{w}$ specifies the best linear discriminant vector.

Algorithm 20.1 shows the pseudo-code for linear discriminant analysis. Here, we assume that there are two classes, and that $\mathbf{S}$ is nonsingular (i.e., $\mathbf{S}^{-1}$ exists). The vector $\mathbf{1}_{n_i}$ is the vector of all ones, with the appropriate dimension for each class, i.e., $\mathbf{1}_{n_i} \in \mathbb{R}^{n_i}$ for class $i = 1, 2$. After dividing $\mathbf{D}$ into the two groups $\mathbf{D}_1$ and $\mathbf{D}_2$, LDA proceeds to compute the between-class and within-class scatter matrices, $\mathbf{B}$ and $\mathbf{S}$. The optimal LD vector is obtained as the dominant eigenvector of $\mathbf{S}^{-1}\mathbf{B}$. In terms of computational complexity, computing $\mathbf{S}$ takes $O(nd^2)$ time, and computing the dominant eigenvalue-eigenvector pair takes $O(d^3)$ time in the worst case. Thus, the total time is $O(d^3 + nd^2)$.

ALGORITHM 20.1. Linear Discriminant Analysis

```
LINEARDISCRIMINANT (D = {(x_i, y_i)}_{i=1}^n):
1 D_i ← {x_j | y_j = c_i, j = 1,...,n}, i = 1,2 // class-specific subsets
2 μ_i ← mean(D_i), i = 1,2 // class means
3 B ← (μ_1 − μ_2)(μ_1 − μ_2)^T // between-class scatter matrix
4 Z_i ← D_i − 1_{n_i} μ_i^T, i = 1,2 // center class matrices
5 S_i ← Z_i^T Z_i, i = 1,2 // class scatter matrices
6 S ← S_1 + S_2 // within-class scatter matrix
7 λ_1, w ← eigen(S^{-1}B) // compute dominant eigenvector
```

Example 20.2 (Linear Discriminant Analysis). Consider the 2-dimensional Iris data (with attributes `sepal length` and `sepal width`) shown in Example 20.1. Class c_1, corresponding to `iris-setosa`, has $n_1 = 50$ points, whereas the other class c_2 has $n_2 = 100$ points. The means for the two classes c_1 and c_2, and their difference is given as

$$\boldsymbol{\mu}_1 = \begin{pmatrix} 5.01 \\ 3.42 \end{pmatrix}^T \qquad \boldsymbol{\mu}_2 = \begin{pmatrix} 6.26 \\ 2.87 \end{pmatrix}^T \qquad \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 = \begin{pmatrix} -1.256 \\ 0.546 \end{pmatrix}^T$$

The between-class scatter matrix is

$$\mathbf{B} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T = \begin{pmatrix} -1.256 \\ 0.546 \end{pmatrix} \begin{pmatrix} -1.256 & 0.546 \end{pmatrix} = \begin{pmatrix} 1.587 & -0.693 \\ -0.693 & 0.303 \end{pmatrix}$$

and the within-class scatter matrix is

$$\mathbf{S}_1 = \begin{pmatrix} 6.09 & 4.91 \\ 4.91 & 7.11 \end{pmatrix} \qquad \mathbf{S}_2 = \begin{pmatrix} 43.5 & 12.09 \\ 12.09 & 10.96 \end{pmatrix} \qquad \mathbf{S} = \mathbf{S}_1 + \mathbf{S}_2 = \begin{pmatrix} 49.58 & 17.01 \\ 17.01 & 18.08 \end{pmatrix}$$

$\mathbf{S}$ is nonsingular, with its inverse given as

$$\mathbf{S}^{-1} = \begin{pmatrix} 0.0298 & -0.028 \\ -0.028 & 0.0817 \end{pmatrix}$$

Therefore, we have

$$\mathbf{S}^{-1}\mathbf{B} = \begin{pmatrix} 0.0298 & -0.028 \\ -0.028 & 0.0817 \end{pmatrix} \begin{pmatrix} 1.587 & -0.693 \\ -0.693 & 0.303 \end{pmatrix} = \begin{pmatrix} 0.066 & -0.029 \\ -0.100 & 0.044 \end{pmatrix}$$

The direction of most separation between c_1 and c_2 is the dominant eigenvector corresponding to the largest eigenvalue of the matrix $\mathbf{S}^{-1}\mathbf{B}$. The solution is

$$J(\mathbf{w}) = \lambda_1 = 0.11$$

$$\mathbf{w} = \begin{pmatrix} 0.551 \\ -0.834 \end{pmatrix}$$

Figure 20.2 plots the optimal linear discriminant direction $\mathbf{w}$, translated to the mean of the data. The projected means for the two classes are shown in black. We can

clearly observe that along $\mathbf{w}$ the circles appear together as a group, and are quite well separated from the triangles. Except for one outlying circle corresponding to the point $(4.5, 2.3)^T$, all points in c_1 are perfectly separated from points in c_2.

For the two class scenario, if $\mathbf{S}$ is nonsingular, we can directly solve for $\mathbf{w}$ without computing the eigenvalues and eigenvectors. Note that $\mathbf{B} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T$ is a $d \times d$ rank-one matrix, and thus $\mathbf{Bw}$ must point in the same direction as $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$ because

$$\begin{aligned}\mathbf{Bw} &= \Big((\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\Big)\mathbf{w} \\ &= (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\Big((\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\mathbf{w}\Big) \\ &= b(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\end{aligned}$$

where $b = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)^T\mathbf{w}$ is just a scalar multiplier.

We can then rewrite Eq. (20.9) as

$$\begin{aligned}\mathbf{Bw} &= \lambda\mathbf{Sw} \\ b(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) &= \lambda\mathbf{Sw} \\ \mathbf{w} &= \frac{b}{\lambda}\mathbf{S}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)\end{aligned}$$

Because $\frac{b}{\lambda}$ is just a scalar, we can solve for the best linear discriminant as

$$\mathbf{w} = \mathbf{S}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \tag{20.10}$$

Once the direction $\mathbf{w}$ has been found we can normalize it to be a unit vector. Thus, instead of solving for the eigenvalue/eigenvector, in the two class case, we immediately obtain the direction $\mathbf{w}$ using Eq. (20.10). Intuitively, the direction that maximizes the separation between the classes can be viewed as a linear transformation (by $\mathbf{S}^{-1}$) of the vector joining the two class means $(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$.

Example 20.3. Continuing Example 20.2, we can directly compute $\mathbf{w}$ as follows:

$$\begin{aligned}\mathbf{w} &= \mathbf{S}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2) \\ &= \begin{pmatrix} 0.066 & -0.029 \\ -0.100 & 0.044 \end{pmatrix}\begin{pmatrix} -1.246 \\ 0.546 \end{pmatrix} = \begin{pmatrix} -0.0527 \\ 0.0798 \end{pmatrix}\end{aligned}$$

After normalizing, we have

$$\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\|} = \frac{1}{0.0956}\begin{pmatrix} -0.0527 \\ 0.0798 \end{pmatrix} = \begin{pmatrix} -0.551 \\ 0.834 \end{pmatrix}$$

Note that even though the sign is reversed for $\mathbf{w}$, compared to that in Example 20.2, they represent the same direction; only the scalar multiplier is different.

20.2 KERNEL DISCRIMINANT ANALYSIS

Kernel discriminant analysis, like linear discriminant analysis, tries to find a direction that maximizes the separation between the classes. However, it does so in *feature space* via the use of kernel functions.

Given a dataset $\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i$ is a point in input space and $y_i \in \{c_1, c_2\}$ is the class label, let $\mathbf{D}_i = \{\mathbf{x}_j | y_j = c_i\}$ denote the data subset restricted to class c_i, and let $n_i = |\mathbf{D}_i|$. Further, let $\phi(\mathbf{x}_i)$ denote the corresponding point in feature space, and let K be a kernel function.

The goal of kernel LDA is to find the direction vector $\mathbf{w}$ in feature space that maximizes

$$\max_{\mathbf{w}} \; J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} \tag{20.11}$$

where m_1 and m_2 are the projected means, and s_1^2 and s_2^2 are projected scatter values in feature space. We first show that $\mathbf{w}$ can be expressed as a linear combination of the points in feature space, and then we transform the LDA objective in terms of the kernel matrix.

Optimal LD: Linear Combination of Feature Points

The mean for class c_i in feature space is given as

$$\boldsymbol{\mu}_i^\phi = \frac{1}{n_i} \sum_{\mathbf{x}_j \in \mathbf{D}_i} \phi(\mathbf{x}_j) \tag{20.12}$$

and the covariance matrix for class c_i in feature space is

$$\boldsymbol{\Sigma}_i^\phi = \frac{1}{n_i} \sum_{\mathbf{x}_j \in \mathbf{D}_i} \left(\phi(\mathbf{x}_j) - \boldsymbol{\mu}_i^\phi\right)\left(\phi(\mathbf{x}_j) - \boldsymbol{\mu}_i^\phi\right)^T$$

Using a derivation similar to Eq. (20.2) we obtain an expression for the between-class scatter matrix in feature space

$$\mathbf{B}_\phi = \left(\boldsymbol{\mu}_1^\phi - \boldsymbol{\mu}_2^\phi\right)\left(\boldsymbol{\mu}_1^\phi - \boldsymbol{\mu}_2^\phi\right)^T = \mathbf{d}_\phi \mathbf{d}_\phi^T \tag{20.13}$$

where $\mathbf{d}_\phi = \boldsymbol{\mu}_1^\phi - \boldsymbol{\mu}_2^\phi$ is the difference between the two class mean vectors. Likewise, using Eqs. (20.5) and (20.6) the within-class scatter matrix in feature space is given as

$$\mathbf{S}_\phi = n_1 \boldsymbol{\Sigma}_1^\phi + n_2 \boldsymbol{\Sigma}_2^\phi$$

$\mathbf{S}_\phi$ is a $d \times d$ symmetric, positive semidefinite matrix, where d is the dimensionality of the feature space. From Eq. (20.9), we conclude that the best linear discriminant vector $\mathbf{w}$ in feature space is the dominant eigenvector, which satisfies the expression

$$\left(\mathbf{S}_\phi^{-1} \mathbf{B}_\phi\right)\mathbf{w} = \lambda \mathbf{w} \tag{20.14}$$

where we assume that $\mathbf{S}_\phi$ is non-singular. Let δ_i denote the ith eigenvalue and $\mathbf{u}_i$ the ith eigenvector of $\mathbf{S}_\phi$, for $i = 1, \ldots, d$. The eigen-decomposition of $\mathbf{S}_\phi$ yields $\mathbf{S}_\phi = \mathbf{U}\boldsymbol{\Delta}\mathbf{U}^T$,

with the inverse of $\mathbf{S}_\phi$ given as $\mathbf{S}_\phi^{-1} = \mathbf{U}\mathbf{\Delta}^{-1}\mathbf{U}^T$. Here $\mathbf{U}$ is the matrix whose columns are the eigenvectors of $\mathbf{S}_\phi$ and $\mathbf{\Delta}$ is the diagonal matrix of eigenvalues of $\mathbf{S}_\phi$. The inverse $\mathbf{S}_\phi^{-1}$ can thus be expressed as the spectral sum

$$\mathbf{S}_\phi^{-1} = \sum_{r=1}^{d} \frac{1}{\delta_r} \mathbf{u}_r \mathbf{u}_r^T \tag{20.15}$$

Plugging Eqs. (20.13) and (20.15) into Eq. (20.14), we obtain

$$\lambda \mathbf{w} = \left(\sum_{r=1}^{d} \frac{1}{\delta_r} \mathbf{u}_r \mathbf{u}_r^T \right) \mathbf{d}_\phi \mathbf{d}_\phi^T \mathbf{w} = \sum_{r=1}^{d} \frac{1}{\delta_r} \left(\mathbf{u}_r (\mathbf{u}_r^T \mathbf{d}_\phi)(\mathbf{d}_\phi^T \mathbf{w}) \right) = \sum_{r=1}^{d} b_r \mathbf{u}_r$$

where $b_r = \frac{1}{\delta_r}(\mathbf{u}_r^T \mathbf{d}_\phi)(\mathbf{d}_\phi^T \mathbf{w})$ is a scalar value. Using a derivation similar to that in Eq. (7.32), the rth eigenvector of $\mathbf{S}_\phi$ can be expressed as a linear combination of the feature points, say $\mathbf{u}_r = \sum_{j=1}^{n} c_{rj} \phi(\mathbf{x}_j)$, where c_{rj} is a scalar coefficient. Thus, we can rewrite $\mathbf{w}$ as

$$\begin{aligned} \mathbf{w} &= \frac{1}{\lambda} \sum_{r=1}^{d} b_r \left(\sum_{j=1}^{n} c_{rj} \phi(\mathbf{x}_j) \right) \\ &= \sum_{j=1}^{n} \phi(\mathbf{x}_j) \left(\sum_{r=1}^{d} \frac{b_r c_{rj}}{\lambda} \right) \\ &= \sum_{j=1}^{n} a_j \phi(\mathbf{x}_j) \end{aligned}$$

where $a_j = \sum_{r=1}^{d} b_r c_{rj}/\lambda$ is a scalar value for the feature point $\phi(\mathbf{x}_j)$. Therefore, the direction vector $\mathbf{w}$ can be expressed as a linear combination of the points in feature space.

LDA Objective via Kernel Matrix

We now rewrite the kernel LDA objective [Eq. (20.11)] in terms of the kernel matrix. Projecting the mean for class c_i given in Eq. (20.12) onto the LD direction $\mathbf{w}$, we have

$$\begin{aligned} m_i = \mathbf{w}^T \boldsymbol{\mu}_i^\phi &= \left(\sum_{j=1}^{n} a_j \, \phi(\mathbf{x}_j) \right)^T \left(\frac{1}{n_i} \sum_{\mathbf{x}_k \in \mathbf{D}_i} \phi(\mathbf{x}_k) \right) \\ &= \frac{1}{n_i} \sum_{j=1}^{n} \sum_{\mathbf{x}_k \in \mathbf{D}_i} a_j \, \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_k) \\ &= \frac{1}{n_i} \sum_{j=1}^{n} \sum_{\mathbf{x}_k \in \mathbf{D}_i} a_j K(\mathbf{x}_j, \mathbf{x}_k) \\ &= \mathbf{a}^T \mathbf{m}_i \end{aligned} \tag{20.16}$$

where $\mathbf{a} = (a_1, a_2, \ldots, a_n)^T$ is the weight vector, and

$$\mathbf{m}_i = \frac{1}{n_i} \begin{pmatrix} \sum_{\mathbf{x}_k \in \mathbf{D}_i} K(\mathbf{x}_1, \mathbf{x}_k) \\ \sum_{\mathbf{x}_k \in \mathbf{D}_i} K(\mathbf{x}_2, \mathbf{x}_k) \\ \vdots \\ \sum_{\mathbf{x}_k \in \mathbf{D}_i} K(\mathbf{x}_n, \mathbf{x}_k) \end{pmatrix} = \frac{1}{n_i} \mathbf{K}^{c_i} \mathbf{1}_{n_i} \tag{20.17}$$

where $\mathbf{K}^{c_i}$ is the $n \times n_i$ subset of the kernel matrix, restricted to columns belonging to points only in $\mathbf{D}_i$, and $\mathbf{1}_{n_i}$ is the n_i-dimensional vector all of whose entries are one. The n-length vector $\mathbf{m}_i$ thus stores for each point in $\mathbf{D}$ its average kernel value with respect to the points in $\mathbf{D}_i$.

We can rewrite the separation between the projected means in feature space as follows:

$$\begin{aligned} (m_1 - m_2)^2 &= \left(\mathbf{w}^T \boldsymbol{\mu}_1^\phi - \mathbf{w}^T \boldsymbol{\mu}_2^\phi\right)^2 \\ &= \left(\mathbf{a}^T \mathbf{m}_1 - \mathbf{a}^T \mathbf{m}_2\right)^2 \\ &= \mathbf{a}^T (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \mathbf{a} \\ &= \mathbf{a}^T \mathbf{M} \mathbf{a} \end{aligned} \tag{20.18}$$

where $\mathbf{M} = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T$ is the between-class scatter matrix.

We can also compute the projected scatter for each class, s_1^2 and s_2^2, purely in terms of the kernel function, as

$$\begin{aligned} s_1^2 &= \sum_{\mathbf{x}_i \in \mathbf{D}_1} \left\| \mathbf{w}^T \phi(\mathbf{x}_i) - \mathbf{w}^T \boldsymbol{\mu}_1^\phi \right\|^2 \\ &= \sum_{\mathbf{x}_i \in \mathbf{D}_1} \left\| \mathbf{w}^T \phi(\mathbf{x}_i) \right\|^2 - 2 \sum_{\mathbf{x}_i \in \mathbf{D}_1} \mathbf{w}^T \phi(\mathbf{x}_i) \cdot \mathbf{w}^T \boldsymbol{\mu}_1^\phi + \sum_{\mathbf{x}_i \in \mathbf{D}_1} \left\| \mathbf{w}^T \boldsymbol{\mu}_1^\phi \right\|^2 \\ &= \left(\sum_{\mathbf{x}_i \in \mathbf{D}_1} \left\| \sum_{j=1}^{n} a_j \phi(\mathbf{x}_j)^T \phi(\mathbf{x}_i) \right\|^2 \right) - 2 \cdot n_1 \cdot \left\| \mathbf{w}^T \boldsymbol{\mu}_1^\phi \right\|^2 + n_1 \cdot \left\| \mathbf{w}^T \boldsymbol{\mu}_1^\phi \right\|^2 \\ &= \left(\sum_{\mathbf{x}_i \in \mathbf{D}_1} \mathbf{a}^T \mathbf{K}_i \mathbf{K}_i^T \mathbf{a} \right) - n_1 \cdot \mathbf{a}^T \mathbf{m}_1 \mathbf{m}_1^T \mathbf{a} \qquad \text{by using Eq. (20.16)} \\ &= \mathbf{a}^T \left(\left(\sum_{\mathbf{x}_i \in \mathbf{D}_1} \mathbf{K}_i \mathbf{K}_i^T \right) - n_1 \mathbf{m}_1 \mathbf{m}_1^T \right) \mathbf{a} \\ &= \mathbf{a}^T \mathbf{N}_1 \mathbf{a} \end{aligned}$$

where $\mathbf{K}_i$ is the ith column of the kernel matrix, and $\mathbf{N}_1$ is the class scatter matrix for c_1. Let $K(\mathbf{x}_i, \mathbf{x}_j) = K_{ij}$. We can express $\mathbf{N}_1$ more compactly in matrix notation as follows:

$$\begin{aligned} \mathbf{N}_1 &= \left(\sum_{\mathbf{x}_i \in \mathbf{D}_1} \mathbf{K}_i \mathbf{K}_i^T \right) - n_1 \mathbf{m}_1 \mathbf{m}_1^T \\ &= (\mathbf{K}^{c_1}) \left(\mathbf{I}_{n_1} - \frac{1}{n_1} \mathbf{1}_{n_1 \times n_1} \right) (\mathbf{K}^{c_1})^T \end{aligned} \tag{20.19}$$

where $\mathbf{I}_{n_1}$ is the $n_1 \times n_1$ identity matrix and $\mathbf{1}_{n_1 \times n_1}$ is the $n_1 \times n_1$ matrix, all of whose entries are 1's.

In a similar manner we get $s_2^2 = \mathbf{a}^T \mathbf{N}_2 \mathbf{a}$, where

$$\mathbf{N}_2 = (\mathbf{K}^{c_2}) \left(\mathbf{I}_{n_2} - \frac{1}{n_2} \mathbf{1}_{n_2 \times n_2} \right) (\mathbf{K}^{c_2})^T$$

where $\mathbf{I}_{n_2}$ is the $n_2 \times n_2$ identity matrix and $\mathbf{1}_{n_2 \times n_2}$ is the $n_2 \times n_2$ matrix, all of whose entries are 1's.

The sum of projected scatter values is then given as

$$s_1^2 + s_2^2 = \mathbf{a}^T (\mathbf{N}_1 + \mathbf{N}_2) \mathbf{a} = \mathbf{a}^T \mathbf{N} \mathbf{a} \tag{20.20}$$

where $\mathbf{N}$ is the $n \times n$ within-class scatter matrix.

Substituting Eqs. (20.18) and (20.20) in Eq. (20.11), we obtain the kernel LDA maximization condition

$$\max_{\mathbf{w}} J(\mathbf{w}) = \max_{\mathbf{a}} J(\mathbf{a}) = \frac{\mathbf{a}^T \mathbf{M} \mathbf{a}}{\mathbf{a}^T \mathbf{N} \mathbf{a}}$$

Notice how all the terms in the expression above involve only kernel functions. The weight vector $\mathbf{a}$ is the eigenvector corresponding to the largest eigenvalue of the generalized eigenvalue problem:

$$\mathbf{M}\mathbf{a} = \lambda_1 \mathbf{N}\mathbf{a} \tag{20.21}$$

If $\mathbf{N}$ is nonsingular, $\mathbf{a}$ is the dominant eigenvector corresponding to the largest eigenvalue for the system

$$(\mathbf{N}^{-1}\mathbf{M})\mathbf{a} = \lambda_1 \mathbf{a}$$

As in the case of linear discriminant analysis [Eq. (20.10)], when there are only two classes we do not have to solve for the eigenvector because $\mathbf{a}$ can be obtained directly:

$$\mathbf{a} = \mathbf{N}^{-1}(\mathbf{m}_1 - \mathbf{m}_2)$$

Once $\mathbf{a}$ has been obtained, we can normalize $\mathbf{w}$ to be a unit vector by ensuring that

$$\mathbf{w}^T \mathbf{w} = 1, \text{ which implies that}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{n} a_i a_j \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j) = 1, \text{ or}$$

$$\mathbf{a}^T \mathbf{K} \mathbf{a} = 1$$

Put differently, we can ensure that $\mathbf{w}$ is a unit vector if we scale $\mathbf{a}$ by $\frac{1}{\sqrt{\mathbf{a}^T \mathbf{K} \mathbf{a}}}$.

Finally, we can project any point $\mathbf{x}$ onto the discriminant direction, as follows:

$$\mathbf{w}^T \phi(\mathbf{x}) = \sum_{j=1}^{n} a_j \phi(\mathbf{x}_j)^T \phi(\mathbf{x}) = \sum_{j=1}^{n} a_j K(\mathbf{x}_j, \mathbf{x}) \tag{20.22}$$

Algorithm 20.2 shows the pseudo-code for kernel discriminant analysis. The method proceeds by computing the $n \times n$ kernel matrix $\mathbf{K}$, and the $n \times n_i$ class

ALGORITHM 20.2. Kernel Discriminant Analysis

KERNELDISCRIMINANT $(\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n, K)$:

1 $\mathbf{K} \leftarrow \left\{K(\mathbf{x}_i, \mathbf{x}_j)\right\}_{i,j=1,\ldots,n}$ `// compute` $n \times n$ `kernel matrix`

2 $\mathbf{K}^{c_i} \leftarrow \left\{\mathbf{K}(j,k) \mid y_k = c_i, 1 \le j,k \le n\right\}, i = 1,2$ `// class kernel matrix`

3 $\mathbf{m}_i \leftarrow \frac{1}{n_i}\mathbf{K}^{c_i}\mathbf{1}_{n_i}, i = 1,2$ `// class means`

4 $\mathbf{M} \leftarrow (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T$ `// between-class scatter matrix`

5 $\mathbf{N}_i \leftarrow \mathbf{K}^{c_i}(\mathbf{I}_{n_i} - \frac{1}{n_i}\mathbf{1}_{n_i \times n_i})(\mathbf{K}^{c_i})^T, i = 1,2$ `// class scatter matrices`

6 $\mathbf{N} \leftarrow \mathbf{N}_1 + \mathbf{N}_2$ `// within-class scatter matrix`

7 $\lambda_1, \mathbf{a} \leftarrow \text{eigen}(\mathbf{N}^{-1}\mathbf{M})$ `// compute weight vector`

8 $\mathbf{a} \leftarrow \frac{\mathbf{a}}{\sqrt{\mathbf{a}^T\mathbf{K}\mathbf{a}}}$ `// normalize` $\mathbf{w}$ `to be unit vector`

specific kernel matrices $\mathbf{K}^{c_i}$ for each class c_i. After computing the between-class and within-class scatter matrices $\mathbf{M}$ and $\mathbf{N}$, the weight vector $\mathbf{a}$ is obtained as the dominant eigenvector of $\mathbf{N}^{-1}\mathbf{M}$. The last step scales $\mathbf{a}$ so that $\mathbf{w}$ will be normalized to be unit length. The complexity of kernel discriminant analysis is $O(n^3)$, with the dominant steps being the computation of $\mathbf{N}$ and solving for the dominant eigenvector of $\mathbf{N}^{-1}\mathbf{M}$, both of which take $O(n^3)$ time.

Example 20.4 (Kernel Discriminant Analysis). Consider the 2-dimensional Iris dataset comprising the `sepal length` and `sepal width` attributes. Figure 20.3a shows the points projected onto the first two principal components. The points have been divided into two classes: c_1 (circles) corresponds to *iris-virginica* and c_2 (triangles) corresponds to the other two Iris types. Here $n_1 = 50$ and $n_2 = 100$, with a total of $n = 150$ points.

Because c_1 is surrounded by points in c_2 a good linear discriminant will not be found. Instead, we apply kernel discriminant analysis using the homogeneous quadratic kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j)^2$$

Solving for $\mathbf{a}$ via Eq. (20.21) yields

$$\lambda_1 = 0.0511$$

However, we do not show $\mathbf{a}$ because it lies in $\mathbb{R}^{150}$. Figure 20.3a shows the contours of constant projections onto the best kernel discriminant. The contours are obtained by solving Eq. (20.22), that is, by solving $\mathbf{w}^T\phi(\mathbf{x}) = \sum_{j=1}^n a_j K(\mathbf{x}_j, \mathbf{x}) = c$ for different values of the scalars c. The contours are hyperbolic, and thus form pairs starting from the center. For instance, the first curve on the left and right of the origin $(0,0)^T$ forms the same contour, that is, points along both the curves have the same value when projected onto $\mathbf{w}$. We can see that contours or pairs of curves starting with the fourth curve (on the left and right) from the center all relate to class c_2, whereas the first three contours deal mainly with class c_1, indicating good discrimination with the homogeneous quadratic kernel.

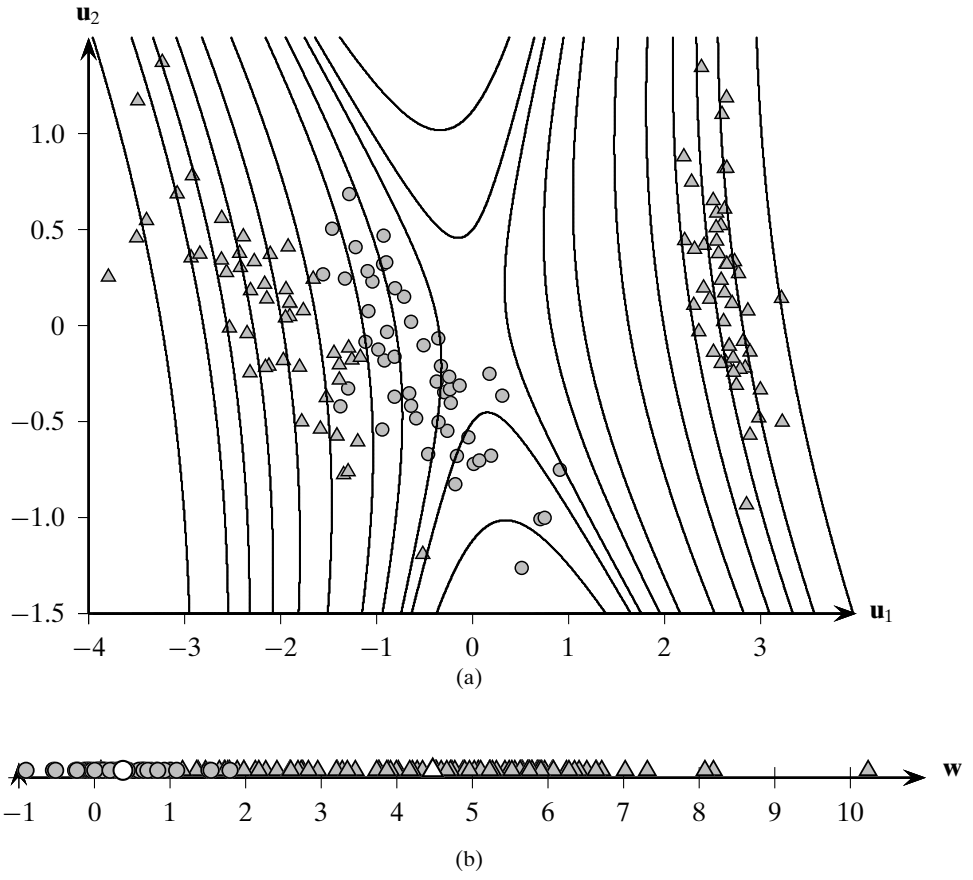

Figure 20.3. Kernel discriminant analysis: quadratic homogeneous kernel.

A better picture emerges when we plot the coordinates of all the points $\mathbf{x}_i \in \mathbf{D}$ when projected onto $\mathbf{w}$, as shown in Figure 20.3b. We can observe that $\mathbf{w}$ is able to separate the two classes reasonably well; all the circles (c_1) are concentrated on the left, whereas the triangles (c_2) are spread out on the right. The projected means are shown in white. The scatters and means for both classes after projection are as follows:

$$m_1 = 0.338 \qquad m_2 = 4.476$$
$$s_1^2 = 13.862 \qquad s_2^2 = 320.934$$

The value of $J(\mathbf{w})$ is given as

$$J(\mathbf{w}) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} = \frac{(0.338 - 4.476)^2}{13.862 + 320.934} = \frac{17.123}{334.796} = 0.0511$$

which, as expected, matches $\lambda_1 = 0.0511$ from above.

In general, it is not desirable or possible to obtain an explicit discriminant vector $\mathbf{w}$, since it lies in feature space. However, because each point $\mathbf{x} = (x_1, x_2)^T \in \mathbb{R}^2$ in

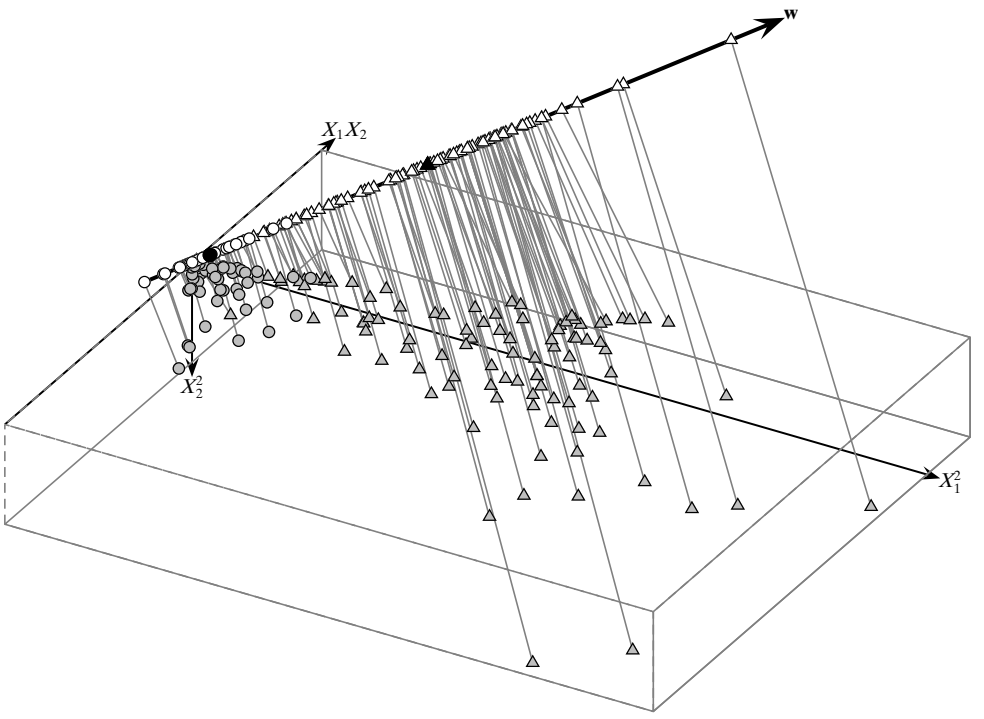

Figure 20.4. Homogeneous quadratic kernel feature space.

input space is mapped to the point $\phi(\mathbf{x}) = (\sqrt{2}x_1x_2, x_1^2, x_2^2)^T \in \mathbb{R}^3$ in feature space via the homogeneous quadratic kernel, for our example it is possible to visualize the feature space, as illustrated in Figure 20.4. The projection of each point $\phi(\mathbf{x}_i)$ onto the discriminant vector $\mathbf{w}$ is also shown, where

$$\mathbf{w} = 0.511x_1x_2 + 0.761x_1^2 - 0.4x_2^2$$

The projections onto $\mathbf{w}$ are identical to those shown in Figure 20.3b.

20.3 FURTHER READING

Linear discriminant analysis was introduced in Fisher (1936). Its extension to kernel discriminant analysis was proposed in Mika et al. (1999). The 2-class LDA approach can be generalized to $k > 2$ classes by finding the optimal $(k-1)$-dimensional subspace projection that best discriminates between the k classes; see Duda, Hart, and Stork (2012) for details.

Duda, R. O., Hart, P. E., and Stork, D. G. (2012). *Pattern Classification*. New York: Wiley-Interscience.

Fisher, R. A. (1936). "The use of multiple measurements in taxonomic problems." *Annals of Eugenics*, 7 (2): 179–188.

Mika, S., Ratsch, G., Weston, J., Scholkopf, B., and Mullers, K. (1999). "Fisher discriminant analysis with kernels." *In Proceedings of the IEEE Neural Networks for Signal Processing Workshop,* IEEE, pp. 41–48.

20.4 EXERCISES

Q1. Consider the data shown in Table 20.1. Answer the following questions:

(a) Compute $\boldsymbol{\mu}_{+1}$ and $\boldsymbol{\mu}_{-1}$, and $\mathbf{S}_B$, the between-class scatter matrix.

(b) Compute $\mathbf{S}_{+1}$ and $\mathbf{S}_{-1}$, and $\mathbf{S}_W$, the within-class scatter matrix.

(c) Find the best direction $\mathbf{w}$ that discriminates between the classes. Use the fact that the inverse of the matrix $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ is given as $\mathbf{A}^{-1} = \frac{1}{\det(\mathbf{A})} \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$.

(d) Having found the direction $\mathbf{w}$, find the point on $\mathbf{w}$ that best separates the two classes.

Table 20.1. Dataset for Q1

i	$\mathbf{x}_i$	y_i
$\mathbf{x}_1$	(4,2.9)	1
$\mathbf{x}_2$	(3.5,4)	1
$\mathbf{x}_3$	(2.5,1)	−1
$\mathbf{x}_4$	(2,2.1)	−1

Q2. Given the labeled points (from two classes) shown in Figure 20.5, and given that the inverse of the within-class scatter matrix is

$$\begin{pmatrix} 0.056 & -0.029 \\ -0.029 & 0.052 \end{pmatrix}$$

Find the best linear discriminant line $\mathbf{w}$, and sketch it.

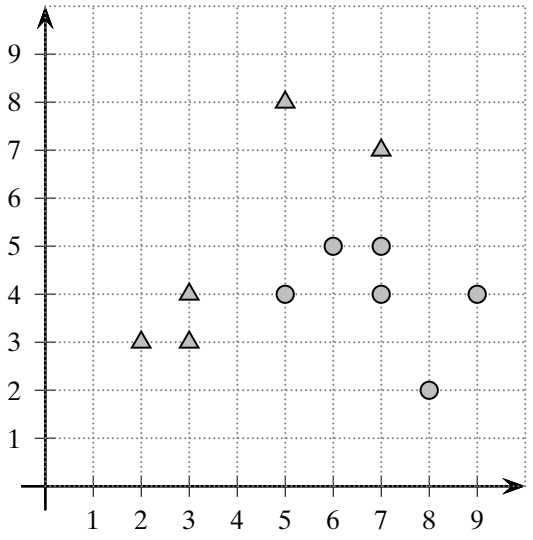

Figure 20.5. Dataset for Q2.

Q3. Maximize the objective in Eq. (20.7) by explicitly considering the constraint $\mathbf{w}^T\mathbf{w} = 1$, that is, by using a Lagrange multiplier for that constraint.

Q4. Prove the equality in Eq. (20.19). That is, show that

$$\mathbf{N}_1 = \left(\sum_{\mathbf{x}_i \in \mathbf{D}_1} \mathbf{K}_i \mathbf{K}_i^T \right) - n_1 \mathbf{m}_1 \mathbf{m}_1^T = (\mathbf{K}^{c_1}) \left(\mathbf{I}_{n_1} - \frac{1}{n_1} \mathbf{1}_{n_1 \times n_1} \right) (\mathbf{K}^{c_1})^T$$