5 *Multivariate Methods*

In chapter 4, we discussed the parametric approach to classification and regression. Now, we generalize this to the multivariate case where we have multiple inputs and where the output, which is class code or continuous output, is a function of these multiple inputs. These inputs may be discrete or numeric. We will see how such functions can be learned from a labeled multivariate sample and also how the complexity of the learner can be fine-tuned to the data at hand.

5.1 Multivariate Data

IN MANY APPLICATIONS, several measurements are made on each individual or event generating an observation vector. The sample may be viewed as a *data matrix*

$$\mathbf{X} = \begin{bmatrix} X_1^1 & X_2^1 & \cdots & X_d^1 \\ X_1^2 & X_2^2 & \cdots & X_d^2 \\ \vdots & & & \\ X_1^N & X_2^N & \cdots & X_d^N \end{bmatrix}$$

where the d columns correspond to d *variables* denoting the result of measurements made on an individual or event. These are also called ***inputs***, ***features***, or ***attributes***. The N rows correspond to independent and identically distributed ***observations***, ***examples***, or ***instances*** on N individuals or events.

For example, in deciding on a loan application, an observation vector is the information associated with a customer and is composed of age, marital status, yearly income, and so forth, and we have N such past customers. These measurements may be of different scales, for example, age in years and yearly income in monetary units. Some like age may be numeric, and some like marital status may be discrete.

Typically these variables are correlated. If they are not, there is no need for a multivariate analysis. Our aim may be *simplification*, that is, summarizing this large body of data by means of relatively few parameters. Or our aim may be *exploratory*, and we may be interested in generating hypotheses about data. In some applications, we are interested in predicting the value of one variable from the values of other variables. If the predicted variable is discrete, this is multivariate classification, and if it is numeric, this is a multivariate regression problem.

5.2 Parameter Estimation

The ***mean vector μ*** is defined such that each of its elements is the mean of one column of **X**:

$$E[\boldsymbol{x}] = \boldsymbol{\mu} = [\mu_1, \ldots, \mu_d]^T \tag{5.1}$$

The variance of X_i is denoted as σ_i^2, and the covariance of two variables X_i and X_j is defined as

$$\sigma_{ij} \equiv \text{Cov}(X_i, X_j) = E[(X_i - \mu_i)(X_j - \mu_j)] = E[X_i X_j] - \mu_i \mu_j \tag{5.2}$$

with $\sigma_{ij} = \sigma_{ji}$, and when $i = j$, $\sigma_{ii} = \sigma_i^2$. With d variables, there are d variances and $d(d - 1)/2$ covariances, which are generally represented as a $d \times d$ matrix, named the ***covariance matrix***, denoted as **Σ**, whose (i, j)th element is σ_{ij}:

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \sigma_{12} & \cdots & \sigma_{1d} \\ \sigma_{21} & \sigma_2^2 & \cdots & \sigma_{2d} \\ \vdots & & & \\ \sigma_{d1} & \sigma_{d2} & \cdots & \sigma_d^2 \end{bmatrix}$$

The diagonal terms are the variances, the off-diagonal terms are the covariances, and the matrix is symmetric. In vector-matrix notation

$$\boldsymbol{\Sigma} \equiv \text{Cov}(\boldsymbol{X}) = E[(\boldsymbol{X} - \boldsymbol{\mu})(\boldsymbol{X} - \boldsymbol{\mu})^T] = E[\boldsymbol{X}\boldsymbol{X}^T] - \boldsymbol{\mu}\boldsymbol{\mu}^T \tag{5.3}$$

If two variables are related in a linear way, then the covariance will be positive or negative depending on whether the relationship has a positive or negative slope. But the size of the relationship is difficult to interpret because it depends on the units in which the two variables are measured. The ***correlation*** between variables $_Xi$ and $_Xj$ is a statistic normalized between -1 and $+1$, defined as

$$\text{Corr}(X_i, X_j) \equiv \rho_{ij} = \frac{\sigma_{ij}}{\sigma_i \sigma_j} \tag{5.4}$$

If two variables are independent, then their covariance, and hence their correlation, is 0. However, the converse is not true: The variables may be dependent (in a nonlinear way), and their correlation may be 0.

Given a multivariate sample, estimates for these parameters can be calculated: The maximum likelihood estimator for the mean is the ***sample mean***, $\boldsymbol{m}$. Its ith dimension is the average of the ith column of $\mathbf{X}$:

$$\boldsymbol{m} = \frac{\sum_{t=1}^{N} \boldsymbol{x}^t}{N} \text{ with } m_i = \frac{\sum_{t=1}^{N} x_i^t}{N}, i = 1, \ldots, d \tag{5.5}$$

The estimator of $\boldsymbol{\Sigma}$ is $\mathbf{S}$, the ***sample covariance*** matrix, with entries

$$s_i^2 = \frac{\sum_{t=1}^{N} (x_i^t - m_i)^2}{N} \tag{5.6}$$

$$s_{ij} = \frac{\sum_{t=1}^{N} (x_i^t - m_i)(x_j^t - m_j)}{N} \tag{5.7}$$

These are biased estimates, but if in an application the estimates vary significantly depending on whether we divide by N or $N - 1$, we are in serious trouble anyway.

The ***sample correlation*** coefficients are

$$r_{ij} = \frac{s_{ij}}{s_i s_j} \tag{5.8}$$

and the sample correlation matrix $\mathbf{R}$ contains r_{ij}.

5.3 Estimation of Missing Values

Frequently, values of certain variables may be missing in observations. The best strategy is to discard those observations all together, but generally we do not have large enough samples to be able to afford this and we do not want to lose data as the non-missing entries do contain information. We try to fill in the missing entries by estimating them. This is called ***imputation***.

In *mean imputation*, for a numeric variable, we substitute the mean (average) of the available data for that variable in the sample. For a discrete variable, we fill in with the most likely value, that is, the value most often seen in the data.

In *imputation by regression*, we try to predict the value of a missing variable from other variables whose values are known for that case. Depending on the type of the missing variable, we define a separate regression or classification problem that we train by the data points for which such values are known. If many different variables are missing, we take the means as the initial estimates and the procedure is iterated until

predicted values stabilize. If the variables are not highly correlated, the regression approach is equivalent to mean imputation.

Depending on the context, however, sometimes the fact that a certain attribute value is missing may be important. For example, in a credit card application, if the applicant does not declare his or her telephone number, that may be a critical piece of information. In such cases, this is represented as a separate value to indicate that the value is missing and is used as such.

5.4 Multivariate Normal Distribution

In the multivariate case where $\boldsymbol{x}$ is d-dimensional and normal distributed, we have

$$p(\boldsymbol{x}) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right] \tag{5.9}$$

and we write $\boldsymbol{x} \sim \mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu}$ is the mean vector and $\boldsymbol{\Sigma}$ is the covariance matrix (see figure 5.1). Just as

$$\frac{(x-\mu)^2}{\sigma^2} = (x-\mu)(\sigma^2)^{-1}(x-\mu)$$

is the squared distance from x to μ in standard deviation units, normalizing for different variances, in the multivariate case the ***Mahalanobis distance*** is used:

$$(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) \tag{5.10}$$

$(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}) = c^2$ is the d-dimensional hyperellipsoid centered at $\boldsymbol{\mu}$, and its shape and orientation are defined by $\boldsymbol{\Sigma}$. Because of the use of the inverse of $\boldsymbol{\Sigma}$, if a variable has a larger variance than another, it receives less weight in the Mahalanobis distance. Similarly, two highly correlated variables do not contribute as much as two less correlated variables. The use of the inverse of the covariance matrix thus has the effect of standardizing all variables to unit variance and eliminating correlations.

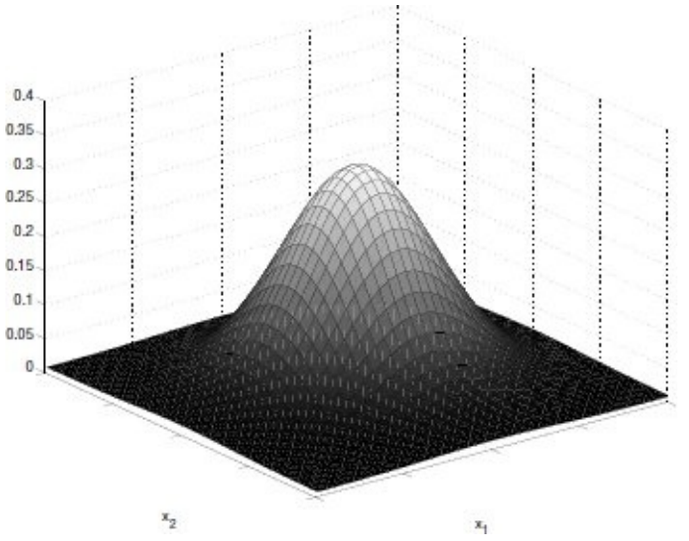

Figure 5.1 Bivariate normal distribution.

Let us consider the bivariate case where $d = 2$ for visualization purposes (see figure 5.2). When the variables are independent, the major axes of the density are parallel to the input axes. The density becomes an ellipse if the variances are different. The density rotates depending on the sign of the covariance (correlation). The mean vector is $\boldsymbol{\mu}^T = [\mu_1, \mu_2]$, and the covariance matrix is usually expressed as

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

The joint bivariate density can be expressed in the form (see exercise 1)

$$p(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} \exp\left[-\frac{1}{2(1-\rho^2)}\left(z_1^2 - 2\rho z_1 z_2 + z_2^2\right)\right] \tag{5.11}$$

where $z_i = (x_i - \mu_i)/\sigma_i$, $i = 1, 2$, are standardized variables; this is called ***z-normalization***. Remember that

$$z_1^2 + 2\rho z_1 z_2 + z_2^2 = \text{constant}$$

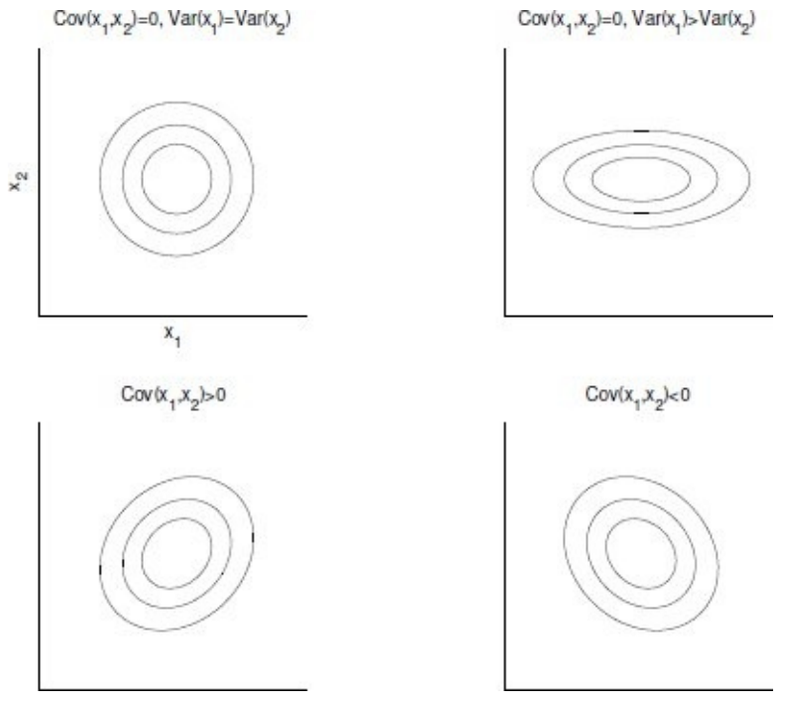

Figure 5.2 Isoprobability contour plot of the bivariate normal distribution. Its center is given by the mean, and its shape and orientation depend on the covariance matrix.

for $|\rho| < 1$, is the equation of an ellipse. When $\rho > 0$, the major axis of the ellipse has a positive slope and if $\rho < 0$, the major axis has a negative slope.

In the expanded Mahalanobis distance of equation 5.11, each variable is normalized to have unit variance, and there is the cross-term that corrects for the correlation between the two variables.

The density depends on five parameters: the two means, the two variances, and the correlation. $\mathbf{\Sigma}$ is nonsingular, and hence positive definite, provided that variances are nonzero and $|\rho| < 1$. If ρ is +1 or −1, the two variables are linearly related, the observations are effectively one-dimensional, and one of the two variables can be disposed of. If $\rho = 0$, then the two variables are independent, the cross-term disappears, and we get a product of two univariate densities.

In the multivariate case, a small value of $|\mathbf{\Sigma}|$ indicates samples are close to $\boldsymbol{\mu}$, just as in the univariate case where a small value of σ^2 indicates samples are close to μ. Small $|\mathbf{\Sigma}|$ may also indicate that there is high correlation between variables. $\mathbf{\Sigma}$ is a symmetric positive definite matrix; this is the multivariate way of saying that $\text{Var}(X) > 0$. If not so, $\mathbf{\Sigma}$ is singular and its determinant is 0. This is either due to linear dependence between the dimensions or because one of the dimensions has variance 0. In such a case, dimensionality should be reduced to a get a positive definite matrix; we discuss methods for this in chapter 6.

If $\boldsymbol{x} \sim \mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then each dimension of $\boldsymbol{x}$ is univariate normal. (The converse is not true: Each X_i may be univariate normal and $\boldsymbol{X}$ may not be multivariate normal.) Actually any $k < d$ subset of the variables is k-variate normal.

A special, *naive* case is where the components of $\boldsymbol{x}$ are independent and $\text{Cov}(X_i, X_j) = 0$, for $i \neq j$, and $\text{Var}(X_i) = \sigma_i^2$, $\forall i$. Then the covariance matrix is diagonal and the joint density is the product of the individual univariate densities:

$$p(\boldsymbol{x}) = \prod_{i-1}^{d} p_i(x_i) = \frac{1}{(2\pi)^{d/2} \prod_{i-1}^{d} \sigma_i} \exp\left[-\frac{1}{2}\sum_{i-1}^{d}\left(\frac{x_i - \mu_i}{\sigma_i}\right)^2\right] \tag{5.12}$$

Now let us see another property we make use of in later chapters. Let us say $\boldsymbol{x} \sim \mathcal{N}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{w} \in \Re^d$, then

$$\boldsymbol{w}^T\boldsymbol{x} = w_1x_1 + w_2x_2 + \cdots + w_dx_d \sim \mathcal{N}(\boldsymbol{w}^T\boldsymbol{\mu}, \boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w})$$

given that

$$\begin{aligned} E[\boldsymbol{w}^T\boldsymbol{x}] &= \boldsymbol{w}^T E[\boldsymbol{x}] = \boldsymbol{w}^T\boldsymbol{\mu} \qquad (5.13)\\ \text{Var}(\boldsymbol{w}^T\boldsymbol{x}) &= E[(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})^2] = E[(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})(\boldsymbol{w}^T\boldsymbol{x} - \boldsymbol{w}^T\boldsymbol{\mu})]\\ &= E[\boldsymbol{w}^T(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T\boldsymbol{w}] = \boldsymbol{w}^T E[(\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T]\boldsymbol{w}\\ &= \boldsymbol{w}^T\boldsymbol{\Sigma}\boldsymbol{w} \qquad (5.14) \end{aligned}$$

That is, the projection of a d-dimensional normal on the vector $\boldsymbol{w}$ is univariate normal. In the general case, if $\mathbf{W}$ is a $d \times k$ matrix with rank $k < d$, then the k-dimensional $\mathbf{W}^T\boldsymbol{x}$ is k-variate normal:

$$\mathbf{W}^T\boldsymbol{x} \sim \mathcal{N}_k(\mathbf{W}^T\boldsymbol{\mu}, \mathbf{W}^T\boldsymbol{\Sigma}\mathbf{W}) \tag{5.15}$$

That is, if we project a d-dimensional normal distribution to a space that is k-dimensional, then it projects to a k-dimensional normal.

5.5 Multivariate Classification

When $\boldsymbol{x} \in \Re^d$, if the class-conditional densities, $p(\boldsymbol{x}|C_i)$, are taken as normal density, $\mathcal{N}_d(\boldsymbol{\mu_i}, \boldsymbol{\Sigma}_i)$, we have

$$p(\boldsymbol{x}|C_i) = \frac{1}{(2\pi)^{d/2}|\boldsymbol{\Sigma}_i|^{1/2}} \exp\left[-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_i)^T\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_i)\right] \tag{5.16}$$

The main reason for this is its analytical simplicity (Duda, Hart, and Stork 2001). Besides, the normal density is a model for many naturally occurring phenomena in that examples of most classes can be seen as mildly changed

versions of a single prototype, $\boldsymbol{\mu}_i$, and the covariance matrix, $\boldsymbol{\Sigma}_i$, denotes the amount of noise in each variable and the correlations of these noise sources. While real data may not often be exactly multivariate normal, it is a useful approximation. In addition to its mathematical tractability, the model is robust to departures from normality as is shown in many works (e.g., McLachlan 1992). However, one clear requirement is that the sample of a class should form a single group; if there are multiple groups, one should use a mixture model (chapter 7).

Let us say we want to predict the type of a car that a customer would be interested in. Different cars are the classes and $\boldsymbol{x}$ are observable data of customers, for example, age and income. $\boldsymbol{\mu}_i$ is the vector of mean age and income of customers who buy car type i and $\boldsymbol{\Sigma}_i$ is their covariance matrix: σ_{i1}^2 and σ_{i2}^2 are the age and income variances, and σ_{i12} is the covariance of age and income in the group of customers who buy car type i.

When we define the discriminant function as

$$g_i(\boldsymbol{x}) = \log p(\boldsymbol{x}|C_i) + \log P(C_i)$$

and assuming $p(\boldsymbol{x}|C_i) \sim \mathcal{N}_d(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, we have

$$g_i(\boldsymbol{x}) = -\frac{d}{2}\log 2\pi - \frac{1}{2}\log|\Sigma_i| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^T\Sigma_i^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i) + \log P(C_i) \tag{5.17}$$

Given a training sample for $K \geq 2$ classes, $\mathcal{X} = \{\boldsymbol{x}^t, \boldsymbol{r}^t\}$, where $r_i^t = 1$ if $\boldsymbol{x}^t \in C_i$ and 0 otherwise, estimates for the means and covariances are found using maximum likelihood separately for each class:

$$\begin{aligned} \hat{P}(C_i) &= \frac{\sum_t r_i^t}{N} \\ \boldsymbol{m}_i &= \frac{\sum_t r_i^t \boldsymbol{x}^t}{\sum_t r_i^t} \\ \mathbf{S}_i &= \frac{\sum_t r_i^t (\boldsymbol{x}^t - \boldsymbol{m}_i)(\boldsymbol{x}^t - \boldsymbol{m}_i)^T}{\sum_t r_i^t} \end{aligned} \tag{5.18}$$

These are then plugged into the discriminant function to get the estimates for the discriminants. Ignoring the first constant term, we have

$$g_i(\boldsymbol{x}) = -\frac{1}{2}\log|\mathbf{S}_i| - \frac{1}{2}(\boldsymbol{x}-\boldsymbol{m}_i)^T\mathbf{S}_i^{-1}(\boldsymbol{x}-\boldsymbol{m}_i) + \log\hat{P}(C_i) \tag{5.19}$$

Expanding this, we get

$$g_i(\boldsymbol{x}) = -\frac{1}{2}\log|\mathbf{S}_i| - \frac{1}{2}\left(\boldsymbol{x}^T\mathbf{S}_i^{-1}\boldsymbol{x} - 2\boldsymbol{x}^T\mathbf{S}_i^{-1}\boldsymbol{m}_i + \boldsymbol{m}_i^T\mathbf{S}_i^{-1}\boldsymbol{m}_i\right) + \log\hat{P}(C_i)$$

which defines a ***quadratic discriminant*** (see figure 5.3) that can also be

written as

$$g_i(\mathbf{x}) = \mathbf{x}^T \mathbf{W}_i \mathbf{x} + \mathbf{w}_i^T \mathbf{x} + w_{i0} \tag{5.20}$$

where

$$\begin{aligned} \mathbf{W}_i &= -\frac{1}{2}\mathbf{S}_i^{-1} \\ \mathbf{w}_i &= \mathbf{S}_i^{-1}\mathbf{m}_i \\ w_{i0} &= -\frac{1}{2}\mathbf{m}_i^T \mathbf{S}_i^{-1}\mathbf{m}_i - \frac{1}{2}\log|\mathbf{S}_i| + \log \hat{P}(C_i) \end{aligned}$$

The number of parameters to be estimated are $K \cdot d$ for the means and $K \cdot d(d + 1)/2$ for the covariance matrices. When d is large and samples are small, $\mathbf{S}_i$ may be singular and inverses may not exist. Or, $|\mathbf{S}_i|$ may be nonzero but too small, in which case it will be unstable; small changes in $\mathbf{S}_i$ will cause large changes in $\mathbf{S}_i^{-1}$. For the estimates to be reliable on small samples, one may want to decrease dimensionality, d, by redesigning the feature extractor and select a subset of the features or somehow combine existing features. We discuss such methods in chapter 6.

Another possibility is to pool the data and estimate a common covariance matrix for all classes:

$$\mathbf{S} = \sum_i \hat{P}(C_i)\mathbf{S}_i \tag{5.21}$$

In this case of equal covariance matrices, equation 5.19 reduces to

$$g_i(\mathbf{x}) = -\frac{1}{2}(\mathbf{x} - \mathbf{m}_i)^T \mathbf{S}^{-1}(\mathbf{x} - \mathbf{m}_i) + \log \hat{P}(C_i) \tag{5.22}$$

The number of parameters is $K \cdot d$ for the means and $d(d + 1)/2$ for the shared covariance matrix. If the priors are equal, the optimal decision rule is to assign input to the class whose mean's Mahalanobis distance to the input is the smallest. As before, unequal priors shift the boundary toward the less likely class. Note that in this case, the quadratic term $\mathbf{x}^T\mathbf{S}^{-1}\mathbf{x}$ cancels since it is common in all discriminants, and the decision boundaries are linear, leading to a ***linear discriminant*** (figure 5.4) that can be written as

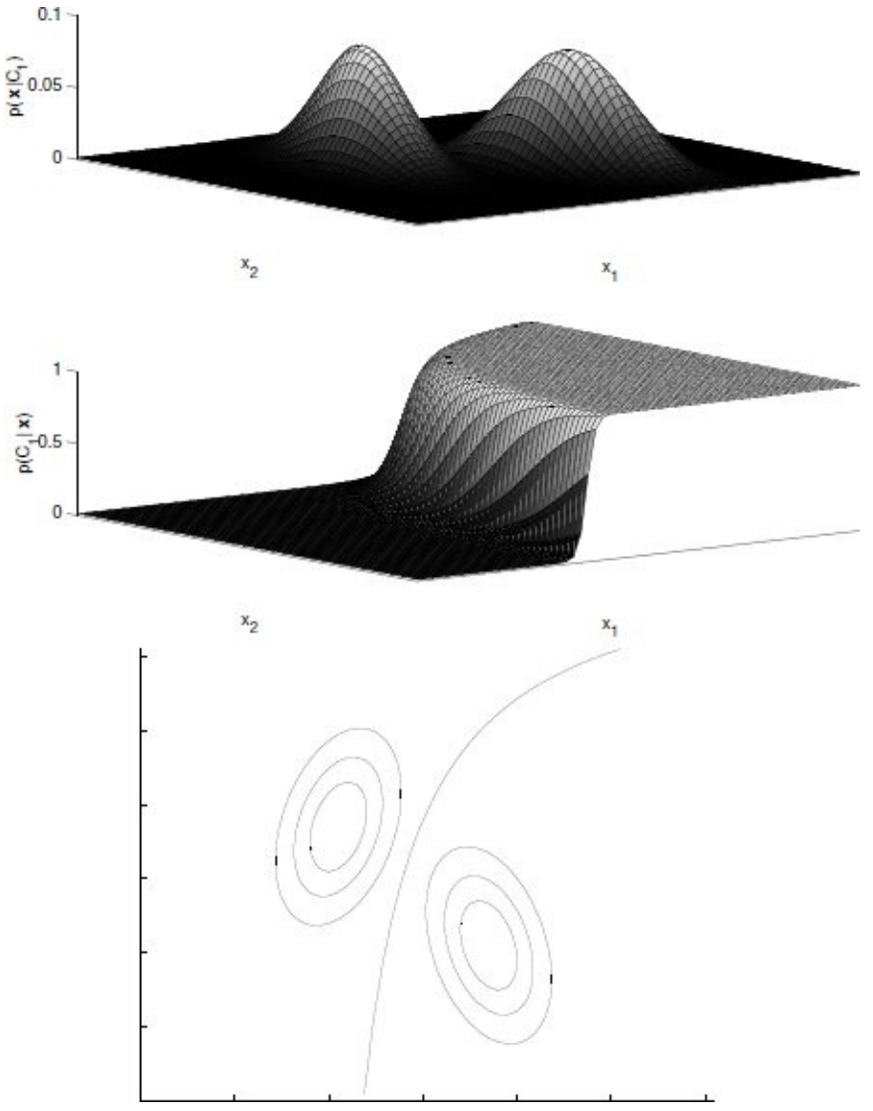

Figure 5.3 Classes have different covariance matrices. Likelihood densities and the posterior probability for one of the classes (top). Class distributions are indicated by isoprobability contours and the discriminant is drawn (bottom).

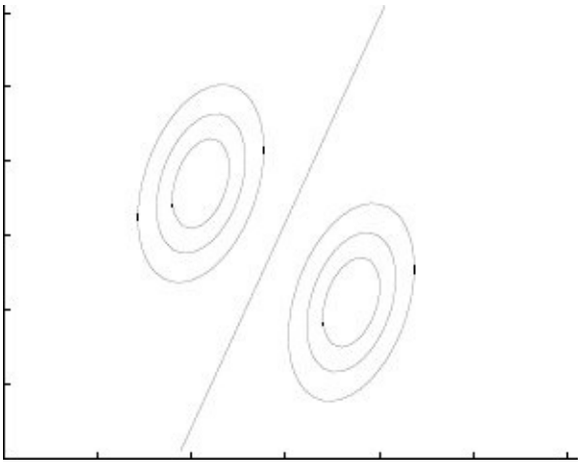

Figure 5.4 Covariances may be arbitary but shared by both classes.

$$g_i(\mathbf{x}) = \mathbf{w}_i^T \mathbf{x} + w_{i0} \tag{5.23}$$

where

$$\begin{aligned} \mathbf{w}_i &= \mathrm{S}^{-1}\mathbf{m}_i \\ w_{i0} &= -\frac{1}{2}\mathbf{m}_i^T \mathrm{S}^{-1}\mathbf{m}_i + \log \hat{P}(C_i) \end{aligned}$$

Decision regions of such a linear classifier are convex; namely, when two points are chosen arbitrarily in one decision region and are connected by a straight line, all the points on the line will lie in the region.

Further simplication may be possible by assuming all off-diagonals of the covariance matrix to be 0, thus assuming independent variables. This is the ***naive Bayes' classifier*** where $p(x_j|C_i)$ are univariate Gaussian. $\mathbf{S}$ and its inverse are diagonal, and we get

$$g_i(\mathbf{x}) = -\frac{1}{2}\sum_{j-1}^{d}\left(\frac{x_j^t - m_{ij}}{s_j}\right)^2 + \log \hat{P}(C_i) \tag{5.24}$$

The term $(x_j^t - m_{ij})/s_j$ has the effect of normalization and measures the distance in terms of standard deviation units. Geometrically speaking, classes are hyperellipsoidal and, because the covariances are zero, are axis-aligned (see figure 5.5). The number of parameters is $K \cdot d$ for the means and d for the variances. Thus the complexity of $\mathbf{S}$ is reduced from $O(d^2)$ to $O(d)$.

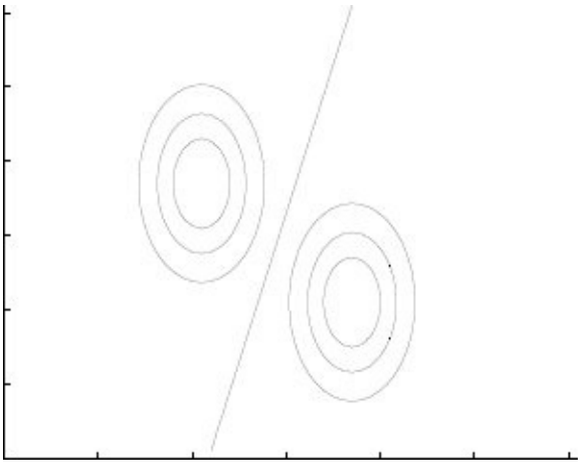

Figure 5.5 All classes have equal, diagonal covariance matrices, but variances are not equal.

Simplifying even further, if we assume all variances to be equal, the Mahalanobis distance reduces to ***Euclidean distance***. Geometrically, the distribution is shaped spherically, centered around the mean vector $\boldsymbol{m}_i$ (see figure 5.6). Then $|\mathbf{S}| = s^{2d}$ and $\mathbf{S}^{-1} = (1/s^2)\mathbf{I}$. The number of parameters in this case is $K \cdot d$ for the means and 1 for s^2.

$$g_i(\boldsymbol{x}) = -\frac{\|\boldsymbol{x} - \boldsymbol{m}_i\|^2}{2s^2} + \log \hat{P}(C_i) = -\frac{1}{2s^2}\sum_{j-1}^{d}(x_j^t - m_{ij})^2 + \log \hat{P}(C_i) \tag{5.25}$$

If the priors are equal, we have $g_i(\boldsymbol{x}) = -\|\boldsymbol{x} - \boldsymbol{m}_i\|^2$. This is named the ***nearest mean classifier*** because it assigns the input to the class of the nearest mean. If each mean is thought of as the ideal prototype or template for the class, this is a ***template matching*** procedure. This can be expanded as

$$\begin{aligned} g_i(\boldsymbol{x}) &= -\|\boldsymbol{x} - \boldsymbol{m}_i\|^2 = -(\boldsymbol{x} - \boldsymbol{m}_i)^T(\boldsymbol{x} - \boldsymbol{m}_i) \\ &= -(\boldsymbol{x}^T\boldsymbol{x} - 2\boldsymbol{m}_i^T\boldsymbol{x} + \boldsymbol{m}_i^T\boldsymbol{m}_i) \end{aligned} \tag{5.26}$$

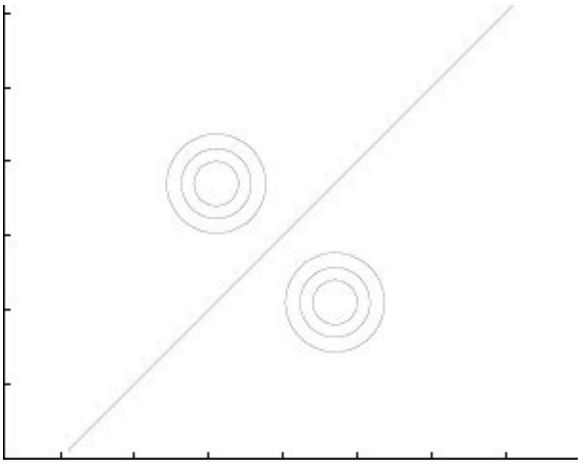

Figure 5.6 All classes have equal, diagonal covariance matrices of equal variances on both dimensions.

The first term, $\boldsymbol{x}^T\boldsymbol{x}$, is shared in all $g_i(\boldsymbol{x})$ and can be dropped, and we can write the discriminant function as

$$g_i(\boldsymbol{x}) = \boldsymbol{w}_i^T\boldsymbol{x} + w_{i0} \tag{5.27}$$

where $\boldsymbol{w}_i = \boldsymbol{m}_i$ and $w_{i0} = -(1/2)\|\boldsymbol{m}_i\|^2$. If all $\boldsymbol{m}_i$ have similar norms, then this term can also be ignored and we can use

$$g_i(\boldsymbol{x}) = \boldsymbol{m}_i^T\boldsymbol{x} \tag{5.28}$$

When the norms of $\boldsymbol{m}_i$ are comparable, dot product can also be used as the similarity measure instead of the (negative) Euclidean distance.

We can actually think of finding the best discriminant function as the task of finding the best distance function. This can be seen as another approach to classification: Instead of learning the discriminant functions, $g_i(\boldsymbol{x})$, we want to learn the suitable distance function $\mathcal{D}(\boldsymbol{x}_1, \boldsymbol{x}_2)$, such that for any $\boldsymbol{x}_1$, $\boldsymbol{x}_2$, $\boldsymbol{x}_3$, where $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ belong to the same class, and $\boldsymbol{x}_1$ and $\boldsymbol{x}_3$ belong to two different classes, we would like to have

$$\mathcal{D}(\boldsymbol{x}_1, \boldsymbol{x}_2) < \mathcal{D}(\boldsymbol{x}_1, \boldsymbol{x}_3)$$

Table 5.1 Reducing variance through simplifying assumptions

Assumption	Covariance matrix	No. of parameters
Shared, Hyperspheric	$\mathbf{S}_i = \mathbf{S} = s^2\mathbf{I}$	1
Shared, Axis-aligned	$\mathbf{S}_i = \mathbf{S}$, with $s_{ij} = 0$	d

Shared, Hyperellipsoidal	$\mathbf{S}_i = \mathbf{S}$	$d(d+1)/2$
Different, Hyperellipsoidal	$\mathbf{S}_i$	$K \cdot (d(d+1)/2)$

5.6 Tuning Complexity

In table 5.1, we see how the number of parameters of the covariance matrix may be reduced, trading off the comfort of a simple model with generality. This is another example of bias/variance dilemma. When we make simplifying assumptions about the covariance matrices and decrease the number of parameters to be estimated, we risk introducing bias (see figure 5.7). On the other hand, if no such assumption is made and the matrices are arbitrary, the quadratic discriminant may have large variance on small datasets. The ideal case depends on the complexity of the problem represented by the data at hand and the amount of data we have. When we have a small dataset, even if the covariance matrices are different, it may be better to assume a shared covariance matrix; a single covariance matrix has fewer parameters and it can be estimated using more data, that is, instances of all classes. This corresponds to using *linear discriminants*, which are very frequently used in classification and which we discuss in more detail in chapter 10.

Note that when we use Euclidean distance to measure similarity, we are assuming that all variables have the same variance and that they are independent. In many cases, this does not hold; for example, age and yearly income are in different units, and are dependent in many contexts. In such a case, the inputs may be separately z-normalized in a preprocessing stage (to have zero mean and unit variance), and then Euclidean distance can be used. On the other hand, sometimes even if the variables are dependent, it may be better to assume that they are independent and to use the naive Bayes' classifier if we do not have enough data to calculate the dependency accurately.

Friedman (1989) proposed a method that combines all these as special cases, named ***regularized discriminant analysis*** (RDA). We remember that regularization corresponds to approaches where one starts with high variance and constrains toward lower variance, at the risk of increasing bias. In the case of parametric classification with Gaussian densities, the covariance matrices can be written as a weighted average of the three special cases:

$$\mathbf{S}_i' = \alpha\sigma^2\mathbf{I} + \beta\mathbf{S} + (1-\alpha-\beta)\mathbf{S}_i \tag{5.29}$$

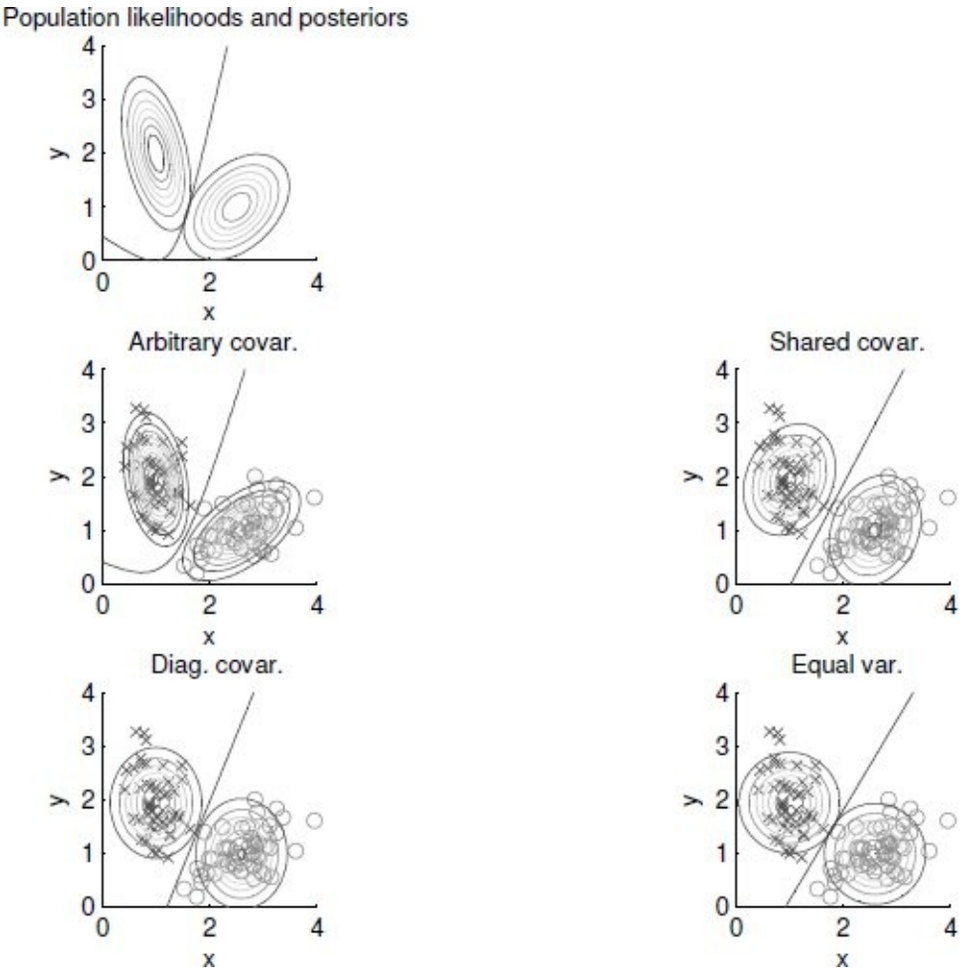

Figure 5.7 Different cases of the covariance matrices fitted to the same data lead to different boundaries.

When $\alpha = \beta = 0$, this leads to a quadratic classifier. When $\alpha = 0$ and $\beta = 1$, the covariance matrices are shared, and we get linear classifiers. When $\alpha = 1$ and $\beta = 0$, the covariance matrices are diagonal with σ^2 on the diagonals, and we get the nearest mean classifier. In between these extremes, we get a whole variety of classifiers where α, β are optimized by cross-validation.

Another approach to regularization, when the dataset is small, is one that uses a Bayesian approach by defining priors on $\boldsymbol{\mu}_i$ and $\mathbf{S}_i$ or that uses cross-validation to choose the best of the four cases given in table 5.1.

5.7 Discrete Features

In some applications, we have discrete attributes taking one of n different values. For example, an attribute may be color $\in$ {red, blue, green, black}, or another may be pixel $\in$ {on, off}. Let us say x_j are binary (Bernoulli) where

$$p_{ij} \equiv p(x_j = 1|C_i)$$

If x_j are independent binary variables, we have

$$p(\mathbf{x}|C_i) = \prod_{j-1}^{d} p_{ij}^{x_j}(1 - p_{ij})^{(1-x_j)}$$

This is another example of the naive Bayes' classifier where $p(x_j|C_i)$ are Bernoulli. The discriminant function is

$$\begin{aligned} g_i(\mathbf{x}) &= \log p(\mathbf{x}|C_i) + \log P(C_i) \\ &= \sum_j \left[x_j \log p_{ij} + (1 - x_j) \log(1 - p_{ij}) \right] + \log P(C_i) \end{aligned} \tag{5.30}$$

which is linear. The estimator for p_{ij} is

$$\hat{p}_{ij} = \frac{\sum_t x_j^t r_i^t}{\sum_t r_i^t} \tag{5.31}$$

This approach is used in ***document categorization***, an example of which is classifying news reports into various categories, such as politics, sports, fashion, and so forth. In the ***bag of words*** representation, we choose a priori d words that we believe give information regarding the class (Manning and Schütze 1999). For example, in news classification, words such as "missile," "athlete," and "couture" are useful, rather than ambiguous words such as "model," or even "runway." In this representation, each text is a d-dimensional binary vector where x_j is 1 if word j occurs in the document and is 0 otherwise. Note that this representation loses all ordering information of words, and hence the name *bag* of words.

After training, $\hat{p}_{ij}$ estimates the probability that word j occurs in document type i. Words whose probabilities are similar for different classes do not convey much information; for them to be useful, we would want the probability to be high for one class (or few) and low for all others; we are going to talk about this type of *feature selection* in chapter 6. Another example application of document categorization is ***spam filtering*** where there are two classes of emails: spam and legitimate. In bioinformatics, too, inputs are generally sequences of discrete items, whether base-pairs or amino acids.

In the general case, instead of binary features, let us say we have the multinomial x_j chosen from the set $\{v_1, v_2, \ldots, v_{n_j}\}$. We define new 0/1 dummy variables as

$$z_{jk}^t = \begin{cases} 1 & \text{if } x_j^t = v_k \\ 0 & \text{otherwise} \end{cases}$$

Let p_{ijk} denote the probability that x_j belonging to C_i takes value v_k:

$$p_{ijk} \equiv p(z_{jk} = 1|C_i) = p(x_j = v_k|C_i)$$

If the attributes are independent, we have

$$p(\mathbf{x}|C_i) = \prod_{j-1}^{d} \prod_{k-1}^{n_j} p_{ijk}^{z_{jk}} \tag{5.32}$$

The discriminant function is then

$$g_i(\mathbf{x}) = \sum_j \sum_k z_{jk} \log p_{ijk} + \log P(C_i) \tag{5.33}$$

The maximum likelihood estimator for p_{ijk} is

$$\hat{p}_{ijk} = \frac{\sum_t z_{jk}^t r_i^t}{\sum_t r_i^t} \tag{5.34}$$

which can be plugged into equation 5.33 to give us the discriminant.

5.8 Multivariate Regression

In ***multivariate linear regression***, the numeric output r is assumed to be written as a linear function, that is, a weighted sum, of several input variables, $x_1, \ldots, x_d$, and noise. Actually in statistical literature, this is called *multiple* regression; statisticians use the term *multivariate* when there are multiple outputs. The multivariate linear model is

$$r^t = g(\mathbf{x}^t|w_0, w_1, \ldots, w_d) + \epsilon = w_0 + w_1 x_1^t + w_2 x_2^t + \cdots + w_d x_d^t + \epsilon \tag{5.35}$$

As in the univariate case, we assume ϵ to be normal with mean 0 and constant variance, and maximizing the likelihood is equivalent to minimizing the sum of squared errors:

$$E(w_0, w_1, \ldots, w_d|X) = \frac{1}{2} \sum_t (r^t - w_0 - w_1 x_1^t - w_2 x_2^t - \cdots - w_d x_d^t)^2 \tag{5.36}$$

Taking the derivative with respect to the parameters, w_j, $j = 0, \ldots, d$, we get these *normal equations*:

$$
\begin{aligned}
\sum_t r^t &= Nw_0 + w_1 \sum_t x_1^t + w_2 \sum_t x_2^t + \cdots + w_d \sum_t x_d^t \\
\sum_t x_1^t r^t &= w_0 \sum_t x_1^t + w_1 \sum_t (x_1^t)^2 + w_2 \sum_t x_1^t x_2^t + \cdots + w_d \sum_t x_1^t x_d^t \\
\sum_t x_2^t r^t &= w_0 \sum_t x_2^t + w_1 \sum_t x_1^t x_2^t + w_2 \sum_t (x_2^t)^2 + \cdots + w_d \sum_t x_2^t x_d^t \\
&\vdots \\
\sum_t x_d^t r^t &= w_0 \sum_t x_d^t + w_1 \sum_t x_d^t x_1^t + w_2 \sum_t x_d^t x_2^t + \cdots + w_d \sum_t (x_d^t)^2
\end{aligned} \tag{5.37}
$$

Let us define the following vectors and matrix:

$$
\mathbf{X} = \begin{bmatrix} 1 & x_1^1 & x_2^1 & \cdots & x_d^1 \\ 1 & x_1^2 & x_2^2 & \cdots & x_d^2 \\ \vdots & & & & \\ 1 & x_1^N & x_2^N & \cdots & x_d^N \end{bmatrix}, \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d, \end{bmatrix}, \mathbf{r} = \begin{bmatrix} r^1 \\ r^2 \\ \vdots \\ r^N \end{bmatrix}
$$

Then the normal equations can be written as

$$
\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{r} \tag{5.38}
$$

and we can solve for the parameters as

$$
\mathbf{w} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{r} \tag{5.39}
$$

This method is the same as we used for polynomial regression using one input. The two problems are the same if we define the variables as $x_1 = x$, $x_2 = x^2$, … , $x_k = x^k$. This also gives us a hint as to how we can do ***multivariate polynomial regression*** if necessary (exercise 10), but unless d is small, in multivariate regression, we rarely use polynomials of an order higher than linear.

Actually using higher-order terms of inputs as additional inputs is only one possibility; we can define any nonlinear function of the original inputs using *basis functions*. For example, we can define new inputs $x_2 = \sin(x)$, $x_3 = \exp(x^2)$ if we believe that such a transformation is useful. Then, using a linear model in this new augmented space will correspond to a nonlinear model in the original space. The same calculation will still be valid; we need only replace $\mathbf{X}$ with the data matrix after the basis functions are applied. As we will see later under various guises (e.g., multilayer perceptrons, support vector machines, Gaussian processes), this type of generalizing the linear model is frequently used.

One advantage of linear models is that after the regression, looking at the w_j, $j = 1, \ldots, d$, values, we can extract knowledge: First, by looking at the signs of w_j, we can see whether x_j have a positive or negative effect on the output. Second, if all x_j are in the same range, by looking at the absolute

values of w_j, we can get an idea about how important a feature is, rank the features in terms of their importances, and even remove the features whose w_j are close to 0.

When there are multiple outputs, this can equivalently be defined as a set of independent single-output regression problems.

5.9 Notes

Appendix B is on linear algebra. For more detail, a good review text on linear algebra is Strang 2006. Harville 1997 is another excellent book that looks at matrix algebra from a statistical point of view.

One inconvenience with multivariate data is that when the number of dimensions is large, one cannot do a visual analysis. There are methods proposed in the statistical literature for displaying multivariate data; a review is given in Rencher 1995. One possibility is to plot variables two by two as bivariate scatter plots: If the data is multivariate normal, then the plot of any two variables should be roughly linear; this can be used as a visual test of multivariate normality. Another possibility that we discuss in chapter 6 is to project them to one or two dimensions and display there.

Most work on pattern recognition is done assuming multivariate normal densities. Sometimes such a discriminant is even called the Bayes' optimal classifier, but this is generally wrong; it is only optimal if the densities are indeed multivariate normal and if we have enough data to calculate the correct parameters from the data. Rencher 1995 discusses tests for assessing multivariate normality as well as tests for checking for equal covariance matrices. McLachlan 1992 discusses classification with multivariate normals and compares linear and quadratic discriminants.

One obvious restriction of multivariate normals is that it does not allow for data where some features are discrete. A variable with n possible values can be converted into n dummy 0/1 variables, but this increases dimensionality. One can do a dimensionality reduction in this n-dimensional space by a method explained in chapter 6 and thereby not increase dimensionality. Parametric classification for such cases of mixed features is discussed in detail in McLachlan 1992.

5.10 Exercises

1. Show equation 5.11.
   SOLUTION: Given that

$$\boldsymbol{\Sigma} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix}$$

we have

$$\begin{aligned} |\boldsymbol{\Sigma}| &= \sigma_1^2\sigma_2^2 - \rho^2\sigma_1^2\sigma_2^2 = \sigma_1^2\sigma_2^2(1-\rho^2) \\ |\boldsymbol{\Sigma}|^{1/2} &= \sigma_1\sigma_2\sqrt{1-\rho^2} \\ \boldsymbol{\Sigma}^{-1} &= \frac{1}{\sigma_1^2\sigma_2^2(1-\rho^2)}\begin{bmatrix} \sigma_2^2 & -\rho\sigma_1\sigma_2 \\ -\rho\sigma_1\sigma_2 & \sigma_1^2 \end{bmatrix} \end{aligned}$$

and $(\boldsymbol{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})$ can be expanded as

$$[x_1-\mu_1\ x_2-\mu_2]\begin{bmatrix} \frac{\sigma_2^2}{\sigma_1^2\sigma_2^2(1-\rho^2)} & -\frac{\rho\sigma_1\sigma_2}{\sigma_1^2\sigma_2^2(1-\rho^2)} \\ -\frac{\rho\sigma_1\sigma_2}{\sigma_1^2\sigma_2^2(1-\rho^2)} & \frac{\sigma_1^2}{\sigma_1^2\sigma_2^2(1-\rho^2)} \end{bmatrix}\begin{bmatrix} x_1-\mu_1 \\ x_2-\mu_2 \end{bmatrix}$$
$$= \frac{1}{1-\rho^2}\left[\left(\frac{x_1-\mu_1}{\sigma_1}\right)^2 - 2\rho\left(\frac{x_1-\mu_1}{\sigma_1}\right)\left(\frac{x_2-\mu_2}{\sigma_2}\right) + \left(\frac{x_2-\mu_2}{\sigma_2}\right)^2\right]$$

2. Generate a sample from a multivariate normal density $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, calculate $\boldsymbol{m}$ and $\mathbf{S}$, and compare them with $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. Check how your estimates change as the sample size changes.
3. Give an example where mean imputation will not work well.

   SOLUTION: Let us say we have two attributes, gender and name. There may be more males than females in the data, but the most frequent name may be "Jane."
4. Generate samples from two multivariate normal densities $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$, $i = 1, 2$, and calculate the Bayes' optimal discriminant for the four cases in table 5.1.
5. For a two-class problem, for the four cases of Gaussian densities in table 5.1, derive

   $$\log\frac{P(C_1|\boldsymbol{x})}{P(C_2|\boldsymbol{x})}$$

6. Another possibility using Gaussian densities is to have them all diagonal but allow them to be different. Derive the discriminant for this case.
7. Let us say in two dimensions, we have two classes with exactly the same mean. What type of boundaries can be defined?
8. Sometimes the data may contain outliers due to noise. How can we find them?
9. In certain applications we can define hierarchies of classes, and this can make discrimination easier. For example, first we discriminate cats from dogs, and then we discriminate between different breeds

of cat. Discuss how this can be done. Can we learn hierarchies from data?

10. Let us say we have two variables x_1 and x_2 and we want to make a quadratic fit using them, namely,

$$f(x_1, x_2) = w_0 + w_1x_1 + w_2x_2 + w_3x_1x_2 + w_4(x_1)^2 + w_5(x_2)^2$$

How can we find w_i, $i = 0, \ldots, 5$, given a sample of $\mathcal{X} = \{x_1^t, x_2^t, r^t\}$?

SOLUTION: We write the fit as

$$f(x_1, x_2) = w_0 + w_1z_1 + w_2z_2 + w_3z_3 + w_4z_4 + w_5z_5$$

where $z_1 = x_1$, $z_2 = x_2$, $z_3 = x_1x_2$, $z_4 = (x_1)^2$, and $z_5 = (x_2)^2$. We can then use linear regression to learn w_i, $i = 0, \ldots, 5$. The linear fit in the five-dimensional $(z_1, z_2, z_3, z_4, z_5)$ space corresponds to a quadratic fit in the two-dimensional (x_1, x_2) space. We discuss such generalized linear models in more detail (and other nonlinear basis functions) in chapter 10.

11. In regression we saw that fitting a quadratic is equivalent to fitting a linear model with an extra input corresponding to the square of the input. Can we also do this in classification?

SOLUTION: Yes. We can define new, auxiliary variables corresponding to powers and cross-product terms and then use a linear model. For example, just as in exercise 10, we can define $z_1 = x_1$, $z_2 = x_2$, $z_3 = x_1x_2$, $z_4 = (x_1)^2$, and $z_5 = (x_2)^2$ and then use a linear model to learn w_i, $i = 0, \ldots, 5$. The linear discriminant in the five-dimensional $(z_1, z_2, z_3, z_4, z_5)$ space corresponds to a quadratic discriminant in the two-dimensional (x_1, x_2) space.

12. In document clustering, ambiguity of words can be decreased by taking the context into account, for example, by considering pairs of words, as in “cocktail party” vs. “party elections.” Discuss how this can be implemented.

5.11 References

Duda, R. O., P. E. Hart, and D. G. Stork. 2001. *Pattern Classification*, 2nd ed. New York: Wiley.

Friedman, J. H. 1989. “Regularized Discriminant Analysis.” *Journal of*

American Statistical Association 84:165–175.

Harville, D. A. 1997. *Matrix Algebra from a Statistician's Perspective*. New York: Springer.

Manning, C. D., and H. Schütze. 1999. *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.

McLachlan, G. J. 1992. *Discriminant Analysis and Statistical Pattern Recognition*. New York: Wiley.

Rencher, A. C. 1995. *Methods of Multivariate Analysis*. New York: Wiley.

Strang, G. 2006. *Linear Algebra and Its Applications*, 4th ed. Boston: Cengage Learning.