Chapter 17
Multivariate Analysis

17.1 Introduction

Much of this book has been devoted to describing relationships among multiple noisy variables, yet we have until now managed to avoid a general discussion of multivariate co-variation. The regression and generalized regression models discussed in Chapters 12, 14, and 15 involved a response variable y that was related to one or more explanatory variables x and this asymmetry of response and explanatory variables allowed us, for the most part, to ignore the co-variation among the whole set of measured variables. In some contexts, however, there are advantages to analyzing multiple measurements together. For instance, in Example 4.7 (p. 100), which involved decoding of wrist movement from MEG signals, the signals came from 87 MEG sensors and it made sense to analyze these collectively, as an 87-dimensional vector at each time point. In this chapter we provide a short overview of methods that have been developed for such purposes, which fall under the heading of *multivariate analysis*, and we return to Example 4.7 on p. 494.

The starting point is the sample mean and sample variance matrix (see Section 4.3.1), while the theory is based largely on the theoretical mean and variance of a random vector (see Section 4.3.1) together with the multivariate normal distribution (see Section 5.5). Section 17.2 reviews the multivariate extensions of t-tests and one-way ANOVA, which are special cases of the general class of methods called *multivariate analysis of variance* (*MANOVA*). MANOVA balances two competing tendencies. On the one hand, when several variables respond similarly to a change in experimental conditions there is stronger evidence for differential response in their combined data than would be provided if each variable were considered separately. This was the idea behind the method of combining p-values from independent tests of the same null hypothesis, described in Section 11.3.1; in Example 11.2 we found that five separate p-values of .02 led to a combined p-value of 2.5×10^{-5}. On the other hand, if the multiple variables are correlated, the assessment must take account of the correlation, and this tends to decrease the effect: in the extreme case of perfect correlation, observing multiple variables becomes the same thing as observing a single

R. E. Kass et al., *Analysis of Neural Data*,
Springer Series in Statistics, DOI: 10.1007/978-1-4614-9602-1_17,

variable. MANOVA incorporates correlation by comparing multivariate co-variation across conditions to that within conditions.

Section 17.3 reviews the main ideas behind dimensionality reduction. When the multiple variables are, collectively, so highly correlated that a variance matrix is no longer of full rank, i.e., no longer positive definite (see p. 618 of the Appendix), some formulas are voided. A solution to this problem is to define a smaller set of new variables that are linear combinations of the original variables, the process of which is called "dimensionality reduction" (though, in general, the combinations do not have to be linear). Dimensionality reduction is also useful for data simplification. For example, data are often displayed by plotting with x and y axes that are suitably defined by a reduction to 2 dimensions.

Section 17.4 returns to the problem of classification, introduced in Section 4.3.4. We first show how Bayes classifiers take a nice form when the classes are defined by multivariate normal distributions, and then go on to describe two commonly-applied alternative methods of classification. In Section 17.4.3 we discuss the concept of *clustering*, which involves putting observations into classes when the classes have not yet been defined and must be estimated or[1] *learned* from the data.

Multivariate analysis uses more advanced mathematics than univariate analysis, and many theoretically-inclined students find in the subject a majestic elegance. While nearly all the methods presented in our synopsis here were developed more than 50 years ago, it is a very active area of continuing research.

17.2 Multivariate Analysis of Variance

17.2.1 MANOVA provides a multivariate extension of ANOVA.

The one-way ANOVA model, given in Eq. (13.1), involves a set of random variables Y_{ij}. We repeat Eq. (13.1) here as Eq. (17.1). The model is

$$Y_{ij} = \mu + \alpha_i + \epsilon_{ij}, \tag{17.1}$$

for $i = 1, \ldots, I$ and $j = 1, \ldots, n_i$ and the usual assumptions are

(i) the ANOVA model (13.1) holds;
(ii) the errors satisfy $E(\epsilon_i) = 0$ for all i;
(iii) the errors ϵ_i are independent of each other;
(iv-1D) $V(\epsilon_i) = \sigma^2$ for all i (homogeneity of error variances), and
(v-1D) $\epsilon_i \sim N(0, \sigma^2)$ (normality of the errors).

[1] The term "learning" tends to be used interchangeably with "estimation," i.e., the process of determining a parameter value from data. Because it may sometimes refer to significance testing, learning is somewhat broader, and it is often associated with techniques used heavily in the field of machine learning. See Hastie et al. (2009).

In Eq. (17.1) each ϵ_{ij} is a random variable, and μ and each α_i are numbers. If we instead take all Y_{ij} and ϵ_{ij} to be p-dimensional random vectors, and μ and all α_i to be vectors, then the model becomes

$$Y_{ij} = \mu + \alpha_i + \epsilon_{ij}, \tag{17.2}$$

which is identical to (17.1) when $p = 1$. The usual assumptions (i–iii) have the same form for (17.2) as for (17.1) while the assumptions we labeled (iv-1D) and (v-1D) become

(iv) $V(\epsilon_i) = \Sigma$ for all i;
(v) $\epsilon_i \sim N(0, \Sigma)$.

Equation (17.2) together with these multivariate assumptions (i–v) then becomes a multivariate analysis of variance (MANOVA) model. Note that in this section we are using Y_{ij} to denote our generic random vector, while in the rest of this chapter we use X.

The idea behind one-way ANOVA is to test the null hypothesis

$$H_0\colon \alpha_i = 0 \tag{17.3}$$

by, first, decomposing the total sum of squares

$$SST = \sum_{i,j} (y_{ij} - \bar{y}_{..})^2 \tag{17.4}$$

using the error sum of squares

$$SSE = \sum_{i,j} (y_{ij} - \bar{y}_{i.})^2 \tag{17.5}$$

as

$$SST = SS_{group} + SSE \tag{17.6}$$

where SS_{group} is defined from (17.6) by subtraction and, second, considering whether[2] SS_{group} is improbably large relative to SSE under H_0. The same idea may be applied in the multivariate case: formulas (17.4) and (17.5) become

$$SST = \sum_{i,j} (y_{ij} - \bar{y}_{..})(y_{ij} - \bar{y}_{..})^T \tag{17.7}$$

and

[2] In constructing the F-statistic, the values of SS_{group} and SSE are first standardized by dividing by their respective degrees of freedom, but that is for the convenience of judging the ratio relative to the number 1.

$$SSE = \sum_{i,j} (y_{ij} - \bar{y}_{i.})(y_{ij} - \bar{y}_{i.})^T \tag{17.8}$$

and then (17.6) may be applied. In addition, under the homogeneity of variance assumption (iv), an estimate of Σ is the *pooled sample variance matrix*

$$S_{pooled} = \frac{1}{N - I} SSE \tag{17.9}$$

where

$$N = \sum_{i=1}^{I} = n_i.$$

On p. 367 we outlined the way the usual one-way ANOVA F-test arises as a likelihood ratio test. This suggests applying a likelihood ratio test in the multivariate setting. The result rejects the null hypothesis of (17.3) when SST is large relative to SSE, where "large" now refers to a matrix and is measured by the determinant (see the appendix, p. 616). Equivalently, the test rejects when the quantity

$$\Lambda = \frac{|SSE|}{|SST|} \tag{17.10}$$

is small. The test was derived by Wilks (1932) and the value Λ is usually called *Wilks' lambda*. An F statistic may be defined in terms of Λ (the expression is not very intuitive; we omit it) and this statistic has, approximately, an F distribution under H_0. The results are usually displayed in a table, much like the ANOVA table given as Table 13.4.

Example 17.1 Functional Specialization of Mouse Visual Areas Because of the potential for genetic manipulation, there is great interest in mouse models of brain function. Cortical areas in the primate visual system can be distinguished according to their differing neural responses. Marshel et al. (2011) sought to provide a similar characterization of mouse visual areas. Specifically, they examined the tuning properties of individual neurons with respect to direction, orientation, spatial frequency, and temporal frequency, across seven visual areas. For each tuning property they devised a measure of sensitivity, yielding a 4-dimensional vector for each neuron. The authors then applied MANOVA to look for differential neural responses in these 4-dimensional vectors across the seven areas. They found the seven areas to be distinguishable using MANOVA, and then proceeded to provide more detailed comparisons for each metric. □

Example 4.7 (continued from p. 100) In their study of decoding wrist movement from MEG sensor recordings, Wang et al. used Bayes classifiers to produce the results in Fig. 4.4. They also evaluated the classification accuracy after averaging the

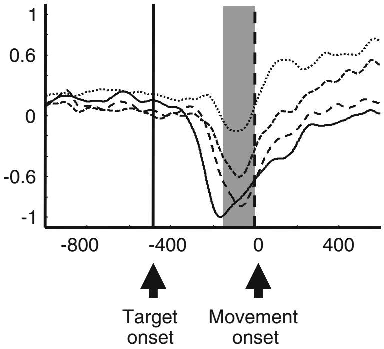

Fig. 17.1 Normalized MEG sensor signals from one subject in the Wang et al. study, averaged across trials. Four traces are shown for a single sensor, corresponding to the four directions of movement. The *shadowed gray region* is the optimal time window found by MANOVA. Adapted from Wang et al. (2010).

sensor recordings across 200 ms time windows. To compute classification accuracy, leave-one-out cross-validation was used. For each subject, and for each trial i, the movement direction on trial i was predicted after the remainder of the trials were used as training data. Using the training data, first an optimal time window for each subject was chosen and then a Bayes classifier was defined (it was assumed that sensor measurements were multivariate normal and the mean and variance parameters were estimated for each of the four directions of movement; see p. 506). The optimal time window of length 200 ms was chosen from 150 possible windows, centered at 150 time points spaced 10 ms apart. To select the optimal time window the authors applied MANOVA in each of the 150 windows, then found the window that produced the largest F statistic. See Fig. 17.1. □

In Section 13.1.3 we said that in the case of two groups, one-way ANOVA reduces to the usual t-test. Similarly, in the case of two groups, MANOVA may be reduced to a simpler form. Let us assume there are n_1 observations in group 1 and n_2 observations in group 2. The pooled sample variance matrix of Eq. (17.9) becomes

$$S_{pooled} = \frac{1}{n_1+n_2-2}\left(\sum_{j=1}^{n_1}(y_{1j}-\bar{y}_1)(y_{1j}-\bar{y}_1)^T + \sum_{j=1}^{n_2}(y_{2j}-\bar{y}_2)(y_{2j}-\bar{y}_2)^T\right) \tag{17.11}$$

which is analogous to the univariate S^2_{pooled} defined in Section 10.3.4. Let us change the notation x used in Section 10.3.4 to y as used here and then write the t-statistic (10.19) in the squared form

$$t^2_{obs} = (\bar{y}_1 - \bar{y}_2)\left((\frac{1}{n_1}+\frac{1}{n_2})s^2_{pooled}\right)^{-1}(\bar{y}_1 - \bar{y}_2). \tag{17.12}$$

The standard test statistic for testing $H_0\colon \alpha_1 - \alpha_2 = 0$ in the multivariate case is

$$T^2 = (\bar{y}_1 - \bar{y}_2)^T \left((\frac{1}{n_1} + \frac{1}{n_2}) S_{pooled} \right)^{-1} (\bar{y}_1 - \bar{y}_2), \tag{17.13}$$

where S_{pooled} is defined above, which is a generalization of (17.12). The statistic T^2 is usually called *Hotelling's* T^2. In this case, under H_0 and the assumptions following Eq. (17.2), including the normality assumption (v), the approximate F distribution of the MANOVA F statistic found by the likelihood ratio test becomes exact and[3] we have

$$\frac{n_1 + n_2 - p}{(n_1 + n_2 - 1)p} T^2 \sim F_{p, n_1 + n_2 - p}.$$

We have discussed one-way MANOVA here, but similar ideas apply to multivariate extensions of two-way ANOVA and more complicated ANOVA designs.

17.2.2 When the variance matrices across conditions are unequal, the likelihood ratio test may be applied.

It sometimes happens that the homogeneity assumption (iv) in the multivariate model (17.2) is violated. The likelihood ratio test may still be used, and p-values may be obtained by simulation.

Example 17.2 Testing Equality of Time-Varying Firing Rates One way to compare the responses of a neuron across two or more experimental conditions is to pick a window of time, compute the spike counts within that window for each of many trials, and then apply a t-test or ANOVA or, possibly, a generalized version of these as in Table 14.7. Sometimes, however, the firing rate may fluctuate across the recorded time interval and it may not be clear what time window would be most appropriate.

Behseta and Kass (2005) and Behseta et al. (2007) suggested, instead, testing the null hypothesis that the firing rate, as a varying function of time, remains the same across the two or more conditions. The situation is illustrated in Fig. 17.2. In the two upper left panels are PSTHs for a motor cortical neuron under two experimental conditions together with smoothed versions of the PSTHs, obtained by methods similar to those of Example 1.1 on p. 422.

The smooth curves in Fig. 17.2 may be considered estimated firing-rate functions, which vary across time. Section 19.3.3 spells this out by defining what is called the *marginal intensity* function $\lambda(t)$ (Eq. (19.23), which is the trial-averaged firing

[3] Here we are using T^2 both as an observed value of a statistic based on data and as a random variable that has a probability distribution. To be consistent with earlier notation, in using T^2 as a random variable we should replace $\bar{y}_1$ and $\bar{y}_2$ in (17.13) and (17.11) with $\bar{Y}_1$ and $\bar{Y}_2$.

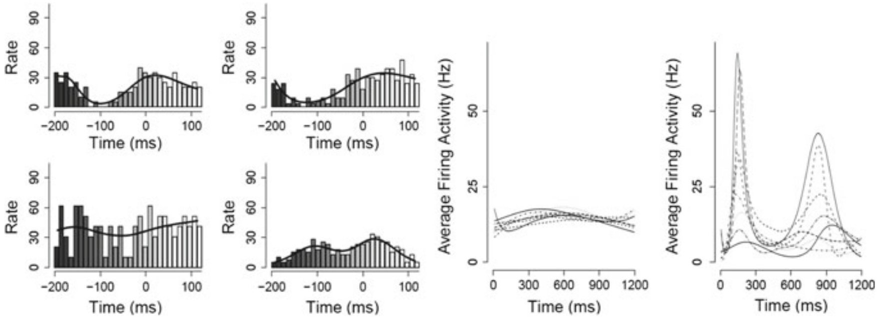

Fig. 17.2 *Left* Responses of two motor cortical neurons. Shown are PSTHs together with *smoothed* versions (*black curves*) obtained from *BARS* (Section 15.2.6). In the two *upper panels* are the estimated firing-rate functions of neuron 1 under two different experimental conditions; for this neuron the firing-rate functions look very similar. In the *lower two panels* are the corresponding estimated firing-rate functions of neuron 2, which look clearly different. Adapted from Behseta and Kass (2005). *Right* Responses of two neurons from the supplementary eye field during eye movements in eight different directions. The first neuron has nearly flat firing-rate functions in all directions, while the second neuron has modulated firing-rate functions which look clearly different. Adapted from Behseta et al. (2007).

rate function (Eq. (19.25)) and, as explained there, the PSTH may be considered an estimate of $\lambda(t)$. To avoid confusion with our use, in this chapter, of λ to denote an eigenvalue, we will here write the trial-averaged firing-rate function instead as $g(t)$. In the case of two firing-rate functions $g_1(t)$ and $g_2(t)$ under two experimental conditions, the null hypothesis becomes $H_0: \ g_1(t) = g_2(t)$ for all t.The smooth curves in the left panels of Fig. 17.2 become estimates $\hat{g}_1(t)$ and $\hat{g}_2(t)$. Behseta and Kass (2005) showed how a version of the T^2 test in (17.13) could be defined from the smooth curves $\hat{g}_1(t)$ and $\hat{g}_2(t)$, together with their estimated variance matrices that come from the smoothing algorithm. As would be expected from Fig. 17.2, the test was not significant for the firing-rate curves in the two upper left panels but was highly significant for the firing-rate curves in the two lower left panels.

Behseta et al. (2007) went on to derive a likelihood ratio test for the more general case in which there are I conditions ($I \geq 2$) and the null hypothesis becomes $H_0 :$ $g_1(t) = g_2(t) = \cdots = g_I(t)$ for all t. This applies to the right-hand panels of Fig. 17.2, which display smoothed firing-rate functions from a pair of supplementary eye field neurons for eye movements in eight directions ($I = 8$). To treat this situation, Behseta et al. 2007 had to allow for the possibility that the variance matrices in each group might be different. Again, the test was not significant for the curves shown for the first neuron but was highly significant for the curves shown for the second neuron. □

17.3 Dimensionality Reduction

17.3.1 A variance matrix may be decomposed into principal components.

The variability of an m-dimensional random vector X is summarized by[4] its variance matrix Σ. According to the spectral decomposition (see p. 617 of the Appendix), we may decompose Σ in the form

$$\Sigma = PDP^T \tag{17.14}$$

where D is an $m \times m$ diagonal matrix and P is an $m \times m$ orthogonal matrix. As discussed on p. 618, the equation $x^T \Sigma x = 1$ defines an m-dimensional ellipse (or *ellipsoid*) the axes of which are defined by the columns of P, which are eigenvectors of Σ. The lengths of these axes are twice the square-root of the corresponding eigenvalues, which are the diagonal elements of D.

Using (12.59) together with the orthogonality relationships $P^T P = PP^T = I_m$, where I_m is the m-dimensional identity matrix, the transformed random vector

$$Y = P^T X \tag{17.15}$$

has variance matrix

$$V(Y) = P^T (PDP^T) P = D. \tag{17.16}$$

Let us assume that the columns of P and diagonal elements of D have been ordered so that $D_{11} \geq D_{22} \geq \cdots \geq D_{mm}$. These diagonal elements, which are eigenvalues of Σ, are usually written $\lambda_j = D_{jj}$, so that

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m.$$

Then, if $\text{col}_j(P)$ is the jth column of P (the jth eigenvector of Σ) the jth component Y_j of Y is given by

$$Y_j = \text{col}_j(P)^T X \tag{17.17}$$

and its variance is

$$V(Y_j) = \text{col}_j(P)^T X = \lambda_j. \tag{17.18}$$

Also, when $i \neq j$, Y_i and Y_j are uncorrelated. If X is multivariate normal, then Y_i and Y_j are independent.

Now for any unit vector u we have

[4] This assumes that the variance matrix is well-defined in the sense that every linear combination $a^T X$ has finite variance. There exist multivariate distributions for which nonzero linear combinations $a^T X$ have infinite variance. We do not consider these here.

$$V(u^T X) \leq \lambda_1. \tag{17.19}$$

Details: Let $w = P^T u$. Notice that

$$w^T w = u^T P P^T u = u^T u = 1$$

so that w is also a unit vector. We compute $V(u^T X)$:

$$V(u^T X) = u^T P D P^T u = w^T D w = \sum_{j=1}^{m} w_j^2 \lambda_j$$

and because $\lambda_j \leq \lambda_1$, we get

$$\sum_{j=1}^{m} w_j^2 \lambda_j \leq \lambda_1 \sum_{j=1}^{m} w_j^2 = \lambda_1.$$

□

Meanwhile, from (17.18) we have that the special case $u = \mathrm{col}_1(P)$ gives

$$V(Y_1) = \lambda_1. \tag{17.20}$$

Together, (17.19) and (17.20) show that Y_1 is the linear combination of components of X that maximizes the variance, among all linear combinations scaled so that the coefficients define a unit vector. In this sense, $\mathrm{col}_j(P)$, the first eigenvector of Σ, defines the *direction of maximal variation* of the random vector X. The linear combination Y_1 is called the *first principal component* of Σ or, more loosely, the first principal component of the distribution of X. Sometimes the term "first principal component" is applied to the first eigenvector $\mathrm{col}_1(P)$.

A similar argument shows that Y_m is the linear combination of components of X that minimizes the variance, among all linear combinations scaled so that the coefficients define a unit vector. With a little more algebra it may also be shown that among all unit vectors u that are perpendicular to $\mathrm{col}_1(P)$, the variance $V(u^T X)$ is maximized by $u = \mathrm{col}_2(P)$. Similarly, $\mathrm{col}_j(P)$ maximizes the variance $V(u^T X)$ among all unit vectors u that are perpendicular to all of $\mathrm{col}_1(P), \mathrm{col}_2(P), \ldots, \mathrm{col}_k(P)$, where $k = j - 1$. The linear combination Y_j is called the jth principal component of Σ.

To summarize, the transformation (17.15), based on the eigenvectors of Σ, produces a new version of X consisting of its principal components. The principal components, given by (17.17), are rotated versions of the components of X that are uncorrelated. If X is multivariate normal, then the principal components are mutually independent. Furthermore, the principal components indicate directions of maximal variation of X in the sense outlined above: the first principal component is in the direction of maximal variation of X, the second principal component is in the direction of maximal variation of X subject to being orthogonal to the first principal component, the third principal component is in the direction of maximal variation of X subject to being orthogonal to the first two principal components, and so on.

Similar analysis may be applied to the sample variance matrix S, defined on p. 90. In this case, we speak of the principal components of S, or of the data vector. This assumes S is of full rank m, i.e. it is positive definite (see p. 617 of the Appendix).

On p. 131 we noted that when a variance matrix Σ is less than full rank, some of its eigenvalues are equal to 0. Suppose there are k positive eigenvalues. Then, as noted on p. 131, Σ may be decomposed instead in terms of the first k eigenvectors, corresponding only to the k positive eigenvalues. These eigenvectors define a k-dimensional subspace in which the variation of X is concentrated. In the case of a sample variance matrix S, which may be considered a noisy estimate of a theoretical variance matrix Σ, the smallest eigenvalues may not be numerically equal to 0 but several may be very close to 0. If we choose a suitable cutoff value c, below which we will say that the smallest eigenvalues are, for practical purposes, the same as 0, then we have effectively determined that there are k positive eigenvalues and the data vector lies in a k-dimensional space. This is the starting point for the idea of dimensionality reduction via principal components: to reduce the dimensionality of a random vector we consider the subspace (the set of linear combinations of its components) corresponding to the positive eigenvalues of its covariance matrix.

Example 17.2 (continued from p. 496) The analysis of Behseta and Kass (2005) involved picking a grid of time values $t_1, \ldots, t_m$ at which to evaluate $\hat{g}_1(t)$ and $\hat{g}_2(t)$. This produced m-dimensional data vectors $(\hat{g}_1(t_1), \ldots, \hat{g}_1(t_m))$ and $(\hat{g}_2(t_1), \ldots, \hat{g}_2(t_m))$ that could be compared based on estimated variance matrices S_1 and S_2 that came from the smoothing method. The authors showed how a statistic similar to T^2 could be defined by replacing the matrix representing the variance of the difference of means, $(\frac{1}{n_1} + \frac{1}{n_2})S_{pooled}$, with $W = S_1 + S_2$, where

$$\begin{aligned} S_1 &= V\left((\hat{g}_1(t_1), \ldots, \hat{g}_1(t_m))\right) \\ S_2 &= V\left((\hat{g}_2(t_1), \ldots, \hat{g}_2(t_m))\right) \end{aligned}$$

which, by independence of the data under the two conditions, satisfies

$$W = V\left((\hat{g}_1(t_1), \ldots, \hat{g}_1(t_m)) - (\hat{g}_2(t_1), \ldots, \hat{g}_2(t_m))\right).$$

Specifically, letting

$$\begin{aligned} U_1 &= (\hat{g}_1(t_1), \ldots, \hat{g}_1(t_m)) \\ U_2 &= (\hat{g}_2(t_1), \ldots, \hat{g}_2(t_m)) \end{aligned}$$

they wished to use a statistic T^2_{curves} given by

$$T^2_{curves} = (U_1 - U_2)^T W^{-1} (U_1 - U_2). \tag{17.21}$$

However, because the grid comprised many time points (m was relatively large), the matrix W was less than full rank, so that (17.21) could not be applied. The authors therefore reduced dimensionality by choosing a suitable small positive number c and retained only the eigenvalues λ_j of W for which $\lambda_j > c$. (The value of c will be discussed below p. 501.) Let us suppose there were k retained eigenvalues, let D_k be the $k \times k$ diagonal matrix having λ_j as its jth diagonal element, and let P_k be the corresponding matrix of the first k eigenvectors of W. Although the matrix $W = PDP^T$ was not of full rank, the $k \times k$ matrix W_k defined by

$$W_k = P_k D_k P_k^T$$

was of full rank k and the new version of the statistic

$$T^2_{curves} = (U_1 - U_2)^T W_k^{-1} (U_1 - U_2)$$

was well-defined. □

The choice of the cutoff c, below which the remaining eigenvalues are treated as equal to 0, is important. As c increases, additional eigenvalues are set to 0 and dimensionality is further reduced. For a given theoretical variance matrix Σ we may identify the eigenvalues that are zero and then consider the subspace corresponding to the positive eigenvalues. But if all we have is a sample variance matrix S, which we view as a noisy estimate of Σ, it may be difficult to determine how many of the corresponding theoretical eigenvalues of Σ are 0. This gives rise to a dramatic extension of the idea of dimensionality reduction: instead of finding a cutoff for which the remaining eigenvalues are nearly 0, the value c could represent a cutoff for which "most of the variation" in the data occurs in the remaining subspace. For this purpose, a standard procedure is to compute the eigenvalues $\hat{\lambda}_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_m$ of S (which are considered to be estimates of $\lambda_1, \lambda_2, \ldots, \lambda_m$) and to declare that the subspace corresponding to the first k eigenvalues *contains a proportion q of the variability* in the data, where q is defined by

$$q = \frac{\hat{\lambda}_1 + \hat{\lambda}_2 + \cdots + \hat{\lambda}_k}{\hat{\lambda}_1 + \hat{\lambda}_2 + \cdots \hat{\lambda}_k + \hat{\lambda}_{k+1} + \cdots + \hat{\lambda}_m}.$$

Data analysts often pick k such that 90 or 95 % of the variability is, in this sense, contained in the subspace defined by the first k principal components.

Example 17.3 Postural Hand Synergies Santello et al. (1998) asked subjects to shape their hand as if grasping and using many familiar objects. The authors defined hand shape using 15 joint angles formed when the subjects were in a static grasp position. The authors reported that roughly 90 % of the variability in these hand shape vectors was accounted for by the first three principal components. They interpreted the 3-dimensional representation to be defined by "synergies," meaning shape combinations resulting from the redundancies in hand movement. □

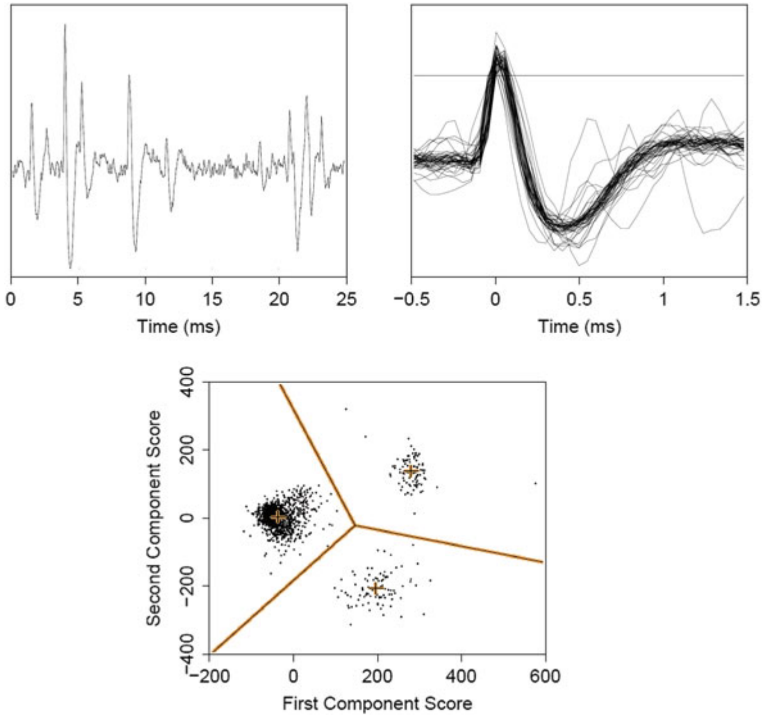

Fig. 17.3 *Top left* A segment of an extracellular electrode voltage recording. *Top right* A plot overlaying the many waveforms from a well-isolated neuron. *Bottom* Three clusters, with cluster boundaries, plotted using axes defined by the first two principal components. The boundaries separating the clusters are defined by $K-$means clustering (see Section 17.4.3). Adapted from Lewicki (1998).

Principal components are also used to visualize data. Let us write the data vectors as $x^1, x^2, \ldots, x^N$. Typically, plots are made of the first two principal components, that is, of the data pairs $(\mathrm{col}_1(P)^T x^i, \mathrm{col}_2(P)^T x^i)$, for $i = 1, \ldots, N$.

Example 17.4 Spike sorting Forebrain Recordings In Example 4.1 we described the problem of spike sorting. Lewicki (1998) reviewed methods and issues and, to illustrate, used a recording from a Zebra finch forebrain. An extracellular electrode records voltage impulses from many different neurons, but each neuron contributes waveforms that are very similar in shape. Several waveforms, apparently from the same neuron, are overlaid in the left panel of Fig. 17.3. Spike sorting attempts to put similar waveforms together into groups or *clusters*, under the assumption that those within a given cluster are likely to emanate from a particular neuron. This poses the statistical machine learning problem of *clustering*, which we discuss in Section 17.4.3.

A spike waveform has a duration of roughly 1.5 ms. If voltage is sampled at 40 kHz (kilohertz) each waveform is a vector of length 60. The data are then all of the waveforms in a recording session, represented as vectors of length 60. Some methods of clustering (including the mixture-of-Gaussians method discussed in Section 17.4.3) have difficulty in high dimensions and it is advantageous to reduce dimensionality. In addition, it can be useful to visualize the data in a two-dimensional space. Principal components may be used for these purposes. The bottom panel of Fig. 17.3 displays a set of the Zebra finch forebrain data plotted using the first two principal components. Three distinct clusters appear, corresponding to waveforms that become identified as coming from three distinct neurons. □

The use of principal components for any purpose is usually called *principal component analysis* (*PCA*).

17.3.2 Methods other than PCA may be used to reduce dimensionality.

Principal component analysis can be very effective in reducing dimensionality of multivariate data that are more-or-less normally distributed. The assumption is that a substantial fraction of the variation lies in a linear subspace, which may be obtained from the principal components corresponding to the large eigenvalues of the variance matrix. Alternatives include methods that attempt to find latent factors, possibly while assuming the data to be non-normal, and methods that assume variation is concentrated in nonlinear subspaces (concentrated in subspaces known[5] as smooth manifolds). We do not discuss methods aimed at finding smooth manifolds on which the variation of X is concentrated, which come under the rubric *manifold learning*. We very briefly describe two other approaches to dimensionality reduction.

The usual *factor analysis* model for an m-dimensional random vector X is given in terms of an $m \times p$ matrix A and a p-dimensional random vector S, with $p < m$, by

$$X = AS + \epsilon$$

where the components of S and ϵ satisfy $S_i \sim N(0, 1)$ and $\epsilon_i \sim N(0, \sigma_i^2)$, all independently, for $i = 1, \ldots, m$. (In this section we are using S to stand for a vector "source" of variation, rather than a sample variance matrix.) The intuition is that the variation of X is driven by a set of p *latent factors*, which are the unobserved (thus, latent, as in Section 16.2) components of S, plus independent noise, and the rows of the matrix A contain the coefficients, called *factor loadings*, that define the combination of factors determining each component of X. Because a fit of the model to data will produce latent factors, and the factor loadings become interpretable, this conception is very appealing. It suffers, however, from a serious difficulty: the

[5] A subspace N of R^m is a smooth manifold if at every point $x \in N$ there is a local coordinate representation in which all points near x in N have the form (u, v) where $v = 0$. In other words, everywhere in N there is a local coordinate system that makes N look like a linear subspace. See Appendix A of Kass and Vos (1997).

unknown parameters are the components of the variance matrix $V(X) = \Sigma$ and for any orthogonal matrix P, if we define $B = AP$, using (12.59) and $PP^T = I_m$ we have

$$\begin{aligned} V(BS + \epsilon) = BV(S)B^T + I_m &= API_mP^TA^T + I_m \\ &= AA^T + I_m \\ &= \Sigma. \end{aligned}$$

In other words, we obtain the same variance matrix using both B and A, so an interpretation of factor loadings based on B would be neither more or less valid than an interpretation based on A. There are thus infinitely many equivalent interpretations. Various methods have been used to specify a unique factor loading matrix, but there often remains a degree of arbitrariness that leaves many practitioners wary of resulting interpretations.[6]

A related, but different approach is to begin by allowing the latent vector S to be non-normal, but with independent components, in the linear latent variable model

$$X = AS,$$

where S and X are both m-dimensional and A is taken to be orthogonal. The idea is that the independent components in S would drive the vector X through the linear combinations in A. If S is assumed to be normally distributed, then so is X, and the solution is given by PCA, i.e., S consists of the principal components. However, if S is allowed to be non-normal it can be quite different.

Let us assume the data vector $X = x$ has been standardized (or *pre-whitened*, see p. 557) so that its sample variance matrix is the m-dimensional identity. We wish to find A and s such that $x = As$. By orthogonality $A^TA = I_m$ so that $A^Tx = s$. The matrix A may be defined to minimize the mutual information among the components of $s = A^Tx$, where mutual information is the Kullback-Leibler divergence between the joint pdf and the independence pdf (estimated from the data), as in (4.28). That is, the components of s are taken to be as close to independent as possible, in the sense of mutual information. The resulting procedure is called *independent components analysis* (*ICA*). It turns out that minimizing mutual information in A^Ts has the effect of making the distribution of s as far from normal as possible (measured in terms of entropy).

Example 17.4 Efficient coding of natural sounds Lewicki (2002) used ICA to find components of auditory signals. Some of the components he found from human speech are shown in Fig. 17.4. For comparison, response properties of cochlear neurons are also displayed. There is a qualitative resemblance between the ICA components and the neural response functions. Lewicki argued that ICA may capture an efficient representation of auditory input. □

[6] The most famous example is Spearman's general intelligence index g, which is obtained from factor analysis. See, e.g., Gould (1996); Devlin et al. (1997).

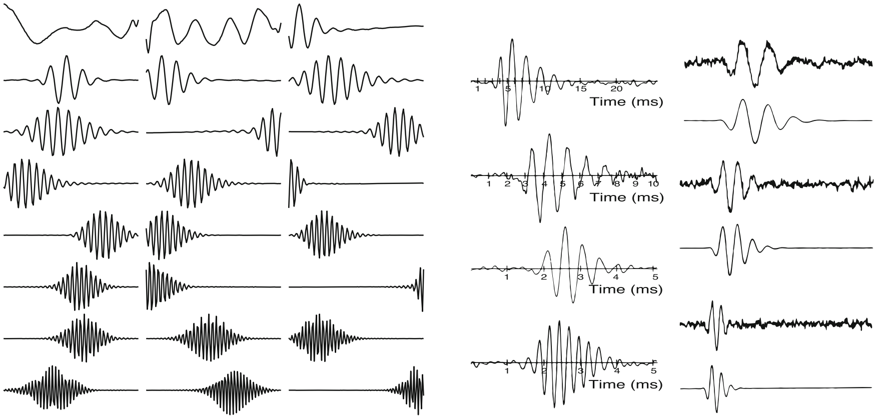

Fig. 17.4 *Left panel* components determined by ICA from human speech. *Right panel* response functions from cochlear neurons. The latter used linear regression of the binary spike train (see Chapter 19) on the input signal at multiple time lags (see p. 530). Adapted from Lewicki (2002).

17.4 Classification and Clustering

17.4.1 Bayes classifiers for multivariate normal distributions take a simple form.

Suppose each of many m-dimensional observation vectors $X = x$ comes from one of K classes $C_1, C_2, \ldots, C_K$, and when it comes from class k the random vector X has pdf $f_k(x)$, for $k = 1, \ldots, K$. The problem of classification (see Section 4.3.4) is to determine, for each observation $X = x$, the class to which x belongs. As we showed in Section 4.3.4, the expected number of classification errors is minimized by using a Bayes classifier. For each x the Bayes classifier finds the class C_k that maximizes the posterior probability given by Eq. (4.38), which we repeat here:

$$P(C = C_k|X = x) = \frac{f_k(x)\pi_k}{\sum_{i=1}^{m} f_i(x)\pi_i}. \tag{17.22}$$

In the special case where, for each class k, we have $X \sim N_m(\mu_k, \Sigma)$ for some μ_k and Σ, the solution takes a simple form. If we write the ratio of posterior probabilities for two classes j and k by plugging the pdfs given by Eq. (5.17) into (17.22), and take logs, after some algebra we obtain

$$\begin{aligned}\log \frac{P(C = C_j|X = x)}{P(C = C_k|X = x)} &= \log \frac{f_j(x)}{f_k(x)} + \log \frac{\pi_j}{\pi_k} \\ &= \delta_j(x) - \delta_k(x) \end{aligned} \tag{17.23}$$

where, for $i = j, k$

$$\delta_i = x^T \Sigma^{-1} \mu_i - \frac{1}{2} \mu_i^T \Sigma^{-1} \mu_i + \log \pi_i. \tag{17.24}$$

In other words, we have $P(C = C_j | X = x) > P(C = C_k | X = x)$ if and only if $\delta_j(x) > \delta_k(x)$, so that the posterior probability is maximized by selecting the class k that maximizes $\delta_i(x)$. The function $\delta_i(x)$ is a linear function of x. It is called the *linear discriminant function*. Classification based on the linear discriminant function is optimal when the classes are defined by multivariate normal distributions all having the same variance matrix.

A similar argument may be applied to the case in which the classes continue to be defined by multivariate normal distributions but the variance matrices are allowed to be different. In this case the linear discriminant functions $\delta_i(x)$ are replaced by quadratic functions of x, which are then called *quadratic discriminant functions*.

In practice, we do not know π_k, μ_k or Σ_k, even when the latter is assumed to satisfy $\Sigma_1 = \cdots = \Sigma_K = \Sigma$. Assuming we have preliminary data arising from known classes from which to *train* the classifier (such data being called *training data*), each prior probability π_k may be estimated by the proportion of training data vectors that fall in class k, i.e., number of training vectors within class k divided by the total number of training data vectors; and we may replace the theoretical means and variance matrices μ_k and Σ_k by the corresponding sample mean and variance calculated within class k. When, for simplicity, it is assumed that $\Sigma_1 = \cdots = \Sigma_K = \Sigma$ the sample variance matrix is pooled across classes as in MANOVA, i.e., the matrix S_{pooled} defined in (17.9) is used, where the groups become the classes. The resulting classification method is called *linear discriminant analysis* (LDA).

Example 4.7 (continued from p. 494) To classify movement directions based on the MEG sensor signals within a 200 ms time window (see Fig. 17.1), Wang et al. used LDA. With this approach the authors reported 4-direction classification accuracies (with chance being 25 %), among nine subjects, ranging from 51.3 to 88.6 % (with a mean of 67 %) for overt movement and 39.6–95 % (with a mean of 62.5 %) for imagined movement. □

LDA often performs well for noisy data, even when the variation is strikingly non-normal. However, for highly structured data alternative methods can do better. See Section 17.4.2.

17.4.2 *Bayes classifiers are not always practical.*

The optimal performance of Bayes classifiers depends on the use of the pdf $f_k(x)$ that generates the m-dimensional random vector X when it comes from class k. In practice, $f_k(x)$ must be estimated from training data which, as m increases, becomes a hard problem unless strong assumptions are made, such as multivariate normality. Even with multivariate normality there are $m(m+1)/2$ parameters to be estimated in the variance matrix Σ, and for large m the data may be insufficient to get good estimates. Sometimes Σ is assumed to be diagonal, so that the components of X become independent. The resulting Bayesian classification procedure is then called *näive Bayes*, which is fast and sometimes effective but it excludes potentially important correlation among the components of X. In general, as the match of the estimated pdfs to the variation in the data deteriorates, the performance of any Bayes classifier may decline. This leads to the problem of designing alternative methods of classification. We describe two popular approaches.

When the data vector satisfies $X \sim N(\mu_k, \Sigma)$ for each class C_k, with $k = 1, \ldots, K$, Eqs. (17.23) and (17.24) give the form of the Bayes classifier in terms of the linear discriminant function. Let us consider, first, the case of binary classification, where $k = 1, 2$. Examining (17.23) and (17.24), if we combine the terms that do not depend on x we may write (17.23) in the alternative form

$$\log \frac{P(C = C_1|X = x)}{P(C = C_2|X = x)} = \alpha_0 + x^T \alpha$$

where $\alpha = \Sigma^{-1}(\mu_1 - \mu_2)$ is an m-dimensional vector. Because, in this binary case, $P(C = C_2|X = x) = 1 - P(C = C_1|X = x)$, we have

$$\log \frac{P(C = C_1|X = x)}{1 - P(C = C_1|X = x)} = \alpha_0 + x^T \alpha. \tag{17.25}$$

Equation (17.25) puts the linear discriminant function in the form of a logistic regression model for binary data, as given by Eq. (14.6), i.e., we could rewrite (17.25) as

$$\log \frac{P(C = C_1|X = x)}{1 - P(C = C_1|X = x)} = \beta_0 + x^T \beta \tag{17.26}$$

and this suggests solving the binary classification problem using logistic regression. More specifically, given training data, the parameters β_0 and β may be estimated using logistic regression applied to the training data to get ML estimates $\hat{\beta}_0$ and $\hat{\beta}$ (as outlined in Section 14.1.2) and then observations may be classified by replacing β_0 and β with $\hat{\beta}_0$ and $\hat{\beta}$ in (17.26) and then assigning an observation to class 1 whenever the function in (17.26) is positive. This method is called a *logistic regression classifier*. The method may be extended to multiple classes using a multi-category generalization of logistic regression, often called *polytomous regression* or *multinomial logistic regression*.

The model in (17.25) looks the same as the model in (17.26) but according to Section 17.4.1, in applying LDA using (17.25) we would estimate the parameters using the sample means and pooled variance matrix. On the other hand, logistic regression would estimate the parameters using maximum likelihood, which is different. The distinction is that logistic regression does not make the assumption of multivariate normality and, instead, treats the x values as fixed.

The general wisdom is that logistic regression classifiers often perform similarly to LDA classifiers. See Hastie et al. (2009) for additional discussion. Although the form of the right-hand side of (17.26) is linear, logistic regression can accommodate complicated nonlinear relationships using the methods discussed in Chapter 15.

A different idea lies behind the *support vector machine* (*SVM*) *classifier*, which we explain briefly by first describing the *perceptron neural network* model. A perceptron model is a function that takes a set of input variables $x_1, \ldots, x_m$ and performs a linear computation followed by binary thresholding:

$$\begin{aligned} \nu &= \phi(u) \\ u &= \left(\sum_{i=1}^{m} w_i x_i \right) - b \end{aligned} \tag{17.27}$$

where $w_1, \ldots, w_m$ are a set of weights associated with that specific perceptron, and $\phi(u) = 1$ when $u \geq 0$ and $\phi(u) = -1$ when $u < 0$. This is a binary classifier in the sense that a vector $x = (x_1, \ldots, x_m)$ is put into class 1 when $\phi(x) = 1$ and into class 2 when $\phi(x) = -1$.

Let us now consider the performance of the perceptron classifier when the data may be separated cleanly into two classes.

Suppose w is an m-dimensional vector. The set $\{x \in R^m : \langle x, w \rangle = 0\}$ is the $(m-1)$-dimensional plane perpendicular to the vector w. It separates two halves of R^m, namely the sets $\{x \in R^m : \langle x, w \rangle > 0\}$ and $\{x \in R^m : \langle x, w \rangle < 0\}$. It is thus called a *separating hyperplane*. The hyperplane $S_0 = \{x \in R^m : \langle x, w \rangle = 0\}$ passes through the origin, i.e., the m-dimensional 0 vector is in this hyperplane ($0 \in S_0$). If $v \in R^m$ we can define $S_v = v + S_0$ to be the set of all vectors in S_0 added to v. This S_v is another separating hyperplane: it may be written

$$S_v = \{x \in R^m : \langle x - v, w \rangle = 0\} = \{x \in R^m : \langle x, w \rangle = b\}$$

where $b = \langle v, w \rangle$ and it separates the sets $\{x \in R^m : \langle x, w \rangle > b\}$ and $\{x \in R^m : \langle x, w \rangle < b\}$.

The separating hyperplane concept applies to data when one set of data vectors lies in a set $\{x \in R^m : \langle x, w \rangle > b\}$ and another set of data lies in a set $\{x \in R^m : \langle x, w \rangle < b\}$. See Fig. 17.5. If two such sets of data come from two distinct classes, then the classifier defined by (17.27) would perfectly classify such data.

The original *perceptron learning rule* attempted to estimate or "learn" the weights $w_1, \ldots, w_m$ from data in order to perform classification. The simple method we have described would be considered ineffective for general-purpose classification, partly because data are not usually perfectly separated in this way and partly because there is not a unique solution: as seen in Fig. 17.5, there are infinitely many separating hyperplanes that fall in the shaded region.

Both of these problems are overcome by classifiers known as *support vector machines* (*SVMs*). Lack of uniqueness is solved by finding the separating hyperplane that maximizes the distance to the closest point in each class. This is found in terms

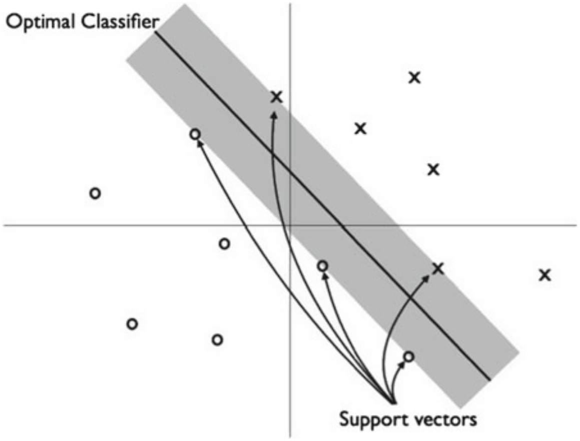

Fig. 17.5 Optimal classification boundary and support vectors for a problem with separable classes. Hypothetical data from two classes are indicated by x and o. The *dark black line* is defined by an optimal classifier that separates the two classes of data. However, any parallel line falling within the gray region would produce the same classification of the given data. The points labeled "support vectors" lie on the boundary of this gray region. The optimal classifier is then determined by maximizing the distance from the separating line to each of the two boundaries of the gray region, which are determined by the support vectors.

of the *support vectors*, which are illustrated in Fig. 17.5. Separation of data vectors is improved by using transformations to higher-dimensional spaces, analogously to what is done in regression when one transforms a single variable x to a polynomial (see Section 12.5.4) or a spline (see Section 15.2). Such transformations take the form $h(x) = (h_1(x), h_2(x), \ldots, h_M(x))$. As the space gets larger, it becomes easier to separate the data vectors from the two classes. One might expect difficulties in implementation, and problems with over-fitting, but there is a so-called *kernel trick* that makes the method[7] practical. It turns out that all of the required computations can be carried out in terms of a *kernel function* $K(u, v)$ that specifies an inner product between m-dimensional vectors u and v,

$$K(u, v) = \langle h(u), h(v) \rangle. \tag{17.28}$$

For example, if we assume $m = 2$, so that $u = (u_1, u_2)$ and $v = (v_1, v_2)$, and we define

$$K(u, v) = (\langle u, v \rangle)^2$$

then (17.28) is satisfied when $h(x)$ (for $x = (x_1, x_2)$) is defined by

$$h(x) = (x_1^2, \sqrt{2}x_1x_2, x_2^2).$$

This simplification allows theory and implementation to be developed.

[7] This use of "kernel" is different than that in Section 15.3.1.

Example 17.5 Predicting Reading Improvement in Dyslexic Children from fMRI To see whether fMRI or diffusion tensor imaging (DTI) might predict future gains in reading ability among dyslexic children, Hoeft et al. (2011) followed 20 such subjects for 2.5 years. The authors split the subjects into two groups based on their improvement in single-word reading skill across the period of observation (high improvement vs. low improvement). They then applied SVM to whole-brain fMRI, and also DTI, to see whether these imaging modalities could be used to predict outcome. They reported 92 % classification accuracy from leave-one-out cross-validation, based on the fMRI data. □

In many situations SVM classifiers behave similarly to logistic regression classifiers, but they are in principle very flexible and sometimes outperform other methods. See Hastie et al. (2009) for additional discussion.

17.4.3 Multivariate observations may be clustered into groups.

In Section 17.4.1 we showed that when a data vector X in class k satisfies $X \sim N_m(\mu_k, \Sigma)$, the Bayes classifier takes the simple form of linear discriminant analysis, given in (17.23) and (17.24). Under the multivariate normality assumption, together with homogeneity of the variance matrices, linear discriminant analysis solves the problem of optimally assigning observations to classes. This, however, requires that the class parameters are known—or that they can be estimated from training data and then treated as known. Estimating parameters from training data is an instance of *supervised learning* because the knowledge of class membership in the training data could be considered a form of supervision. The corresponding *unsupervised* problem of putting data into classes with no prior knowledge of class structure is called *clustering*, and the resulting empirically-defined classes are called *clusters*. We provided an illustration of clustering in Example 17.4 on p. 502.

To discuss the problem in generality, let us assume there are K classes, that X is drawn from class k with probability π_k, and that, conditionally on X being drawn from class k, X follows an m-dimensional multivariate normal distribution with mean μ_k and variance matrix Σ_k. We could write this latter statement as $X|C = k \sim N_m(\mu_k, \Sigma_k)$. We then have a two-stage distribution for X, the first stage involving the distribution of class membership C and the second stage involving the multivariate normal distribution. Taking account of both of these, the marginal distribution of X (after marginalizing over the distribution of C) has pdf found by averaging over C:

$$f(x) = \sum_{k=1}^{K} \pi_k f_k(x; \mu_k, \Sigma_k) \tag{17.29}$$

where $f_k(x; \mu_k, \Sigma_k)$ is the $N_m(\mu_k, \Sigma_k)$ pdf given by (5.17). This is a *mixture model* in the sense that the K multivariate normal distributions are "mixed" according to the prior probabilities $\pi_1, \ldots, \pi_K$. The distribution defined by (17.29) is a *mixture of Gaussians model*, as in the illustration in Section 8.4.5. *Mixture of Gaussians clustering* applies ML estimation to the collection of observations x to estimate the parameters $\mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_K$ and the prior probabilities $\pi_1, \ldots, \pi_K$, and then uses the resulting Bayes classifier to assign each observation x to a cluster. As discussed in Section 8.4.5, ML estimation in mixture of Gaussian models is often implemented using the EM algorithm. Because, in practice, the number of clusters is not known in advance, the model is typically fitted for several different values of K and then a model selection procedure such as AIC or BIC is used (see Section 12.5.7).

In mixture of Gaussians clustering the variance matrices $\Sigma_1, \ldots, \Sigma_K$ must be estimated from the data, and sometimes the data are too sparse to get good estimates of the many variance and covariance parameters. In this case the variance matrices are often assumed[8] equal, $\Sigma_1 = \cdots = \Sigma_K$. A more extreme assumption is to take $\Sigma_1 = \cdots = \Sigma_K = \sigma^2 I_m$ for some σ, i.e., to assume all the variance matrices are equal to a multiple of the m-dimensional identity matrix. This turns out to be closely related to another method, known as K-means clustering.

In K-means clustering it is assumed there are K clusters, with the kth cluster having a mean μ_k. The idea is to put the data vector x into the cluster having its mean closest to x. Thus, after the procedure is applied, so that the clusters are determined and the means μ_k are fixed (by setting them equal to estimated values), every data vector x in cluster j will satisfy

$$||x - \mu_j|| = \min_{k=1,\ldots,K} ||x - \mu_k||. \tag{17.30}$$

However, initially the clusters are not known. They are determined iteratively. After an arbitrary initialization that assigns each data vector to one of K clusters, the following steps are iterated:

1. For $k = 1, \ldots, K$, the mean vectors μ_k is set equal to the sample mean $\bar{x}^k$ of the vectors assigned to cluster k;
2. Each x is assigned to the cluster that minimizes distance as in (17.30).

At each iteration, this algorithm will reduce the sum of squared distances $||x - \mu_j||^2$, summed over all data values, with μ_j being the mean of the cluster to which x is assigned. The algorithm converges to a local minimum of the sum of squared distances (it may not be the global minimum).

Example 17.4 (continued from p. 502) The three clusters in the bottom panel of Fig. 17.3 were identified by K-means clustering (here, with $K = 3$). Three boundary lines are also drawn in Fig. 17.3. Each line is equally distant from the sample means in two of the clusters. □

The relationship of K-means clustering to mixture-of-Gaussian clustering is spelled out in many sources (e.g., Hastie et al. 2009). If it is assumed that $\Sigma_1 =$

[8] Each matrix Σ_k has $m(m+1)/2$ parameters so there are $Km(m+1)/2$ parameters when the matrices are allowed to be different and only $m(m+1)/2$ if they are assumed to be equal.

$\cdots = \Sigma_K = \sigma^2 I_m$ for some σ, and we write the ith data vector as x^i, for $i = 1, \ldots, n$, then the maximum likelihood estimate of μ_k in the mixture-of-Gaussians model, for $k = 1, \ldots, K$, is given by

$$\hat{\mu}_k = \frac{\sum_{i=1}^n \gamma_{ik} x^i}{\sum_{i=1}^n \gamma_{ik}} \tag{17.31}$$

where γ_{ik} is the posterior probability that observation x^i is in class k (see Eq. (8.48)), and is estimated from the data (see p. 217). This is not the same estimate as the sample mean $\bar{x}^k$ over the observations within cluster k. However, when the posterior probabilities become close to 0 and 1 we get

$$\hat{\mu}_k \approx \bar{x}^k.$$

This occurs when the data form highly distinct clusters or, equivalently, when σ is close to 0 relative to the distance between the means of the clusters.