CHAPTER

MAXIMUM LIKELIHOOD ESTIMATION 12

CHAPTER CONTENTS

MLE is a generic method for parameter estimation proposed in the early twentieth century. Thanks to its excellent theoretical and practical properties, it is still one of the most popular techniques even now and it forms the basis of various advanced machine learning techniques. In this chapter, the definition of MLE, its application to Gaussian models, and its usage in pattern recognition are explained.

12.1 DEFINITION

In this section, the definition of MLE is provided.

A set of probability density functions specified by a finite number of parameters is called a *parametric model*. Let us denote a parametric model by $q(\boldsymbol{x};\boldsymbol{\theta})$, a parameter vector by $\boldsymbol{\theta}$, and the domain of parameters by Θ, respectively. Let b be the dimensionality of $\boldsymbol{\theta}$:

$$\boldsymbol{\theta} = (\theta^{(1)},\ldots,\theta^{(b)})^\top.$$

In the notation $q(\boldsymbol{x};\boldsymbol{\theta})$, $\boldsymbol{x}$ before the semicolon is a random variable and $\boldsymbol{\theta}$ after the semicolon is a parameter.

A natural idea to specify the value of parameter $\boldsymbol{\theta}$ is to maximize the chance of obtaining the current training samples $\{\boldsymbol{x}\}_{i=1}^n$. To this end, let us consider the probability that the training samples $\{\boldsymbol{x}\}_{i=1}^n$ are produced under parameter $\boldsymbol{\theta}$. This probability viewed as a function of parameter $\boldsymbol{\theta}$ is called the *likelihood* and denoted by $L(\boldsymbol{\theta})$. Under the i.i.d. assumption (see Section 11.2), the likelihood is expressed as

An Introduction to Statistical Machine Learning. DOI: 10.1016/B978-0-12-802121-7.00023-6

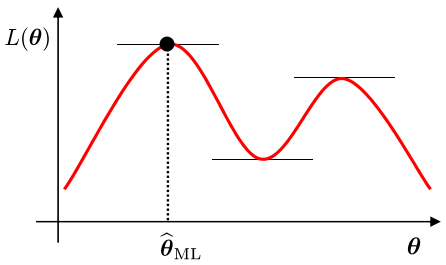


FIGURE 12.1

Likelihood equation, setting the derivative of the likelihood to zero, is a necessary condition for the maximum likelihood solution but is not a sufficient condition in general.

$$L(\boldsymbol{\theta}) = \prod_{i=1}^{n} q(\boldsymbol{x}_i; \boldsymbol{\theta}).$$

MLE finds the maximizer of the likelihood,

$$\widehat{\boldsymbol{\theta}}_{\mathrm{ML}} = \operatorname*{argmax}_{\boldsymbol{\theta} \in \Theta} \; L(\boldsymbol{\theta}),$$

and then a density estimator is given by

$$\widehat{p}(\boldsymbol{x}) = q(\boldsymbol{x}; \widehat{\boldsymbol{\theta}}_{\mathrm{ML}}).$$

If parametric model $q(\boldsymbol{x}; \boldsymbol{\theta})$ is differentiable with respect to $\boldsymbol{\theta}$, $\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}$ satisfies

$$\frac{\partial}{\partial \boldsymbol{\theta}} L(\boldsymbol{\theta}) \bigg|_{\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}}_{\mathrm{ML}}} = \mathbf{0}_b, \tag{12.1}$$

where $\mathbf{0}_b$ denotes the b-dimensional zero vector, and $\frac{\partial}{\partial \boldsymbol{\theta}}$ denotes the *partial derivative* with respect to $\boldsymbol{\theta}$. The partial derivative with respect to vector $\boldsymbol{\theta}$ gives the b-dimensional vector whose ℓth element is given by $\frac{\partial}{\partial \theta^{(\ell)}}$:

$$\frac{\partial}{\partial \boldsymbol{\theta}} = \left(\frac{\partial}{\partial \theta^{(1)}}, \ldots, \frac{\partial}{\partial \theta^{(b)}} \right)^{\top}.$$

Eq. (12.1) is called the *likelihood equation* and is a *necessary condition* for the maximum likelihood solution. Note, however, that it is not generally a *sufficient condition*, i.e., the maximum likelihood solution always satisfies Eq. (12.1), but solving Eq. (12.1) does not necessarily give the solution (Fig. 12.1).

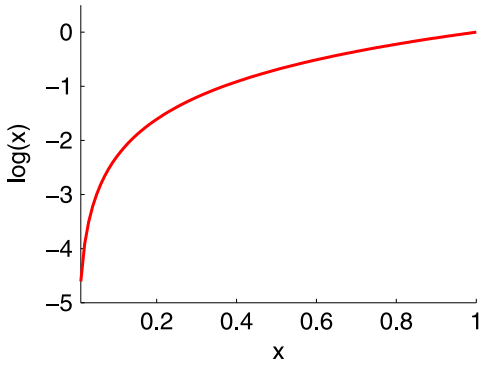


FIGURE 12.2

Log function is monotone increasing.

Since the log function is monotone increasing, the maximizer of the likelihood can also be obtained by maximizing the *log-likelihood* (Fig. 12.2):

$$\widehat{\boldsymbol{\theta}}_{\mathrm{ML}} = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}} \ \log L(\boldsymbol{\theta}) = \underset{\boldsymbol{\theta}\in\Theta}{\operatorname{argmax}} \left[\sum_{i=1}^{n} \log q(\boldsymbol{x}_i;\boldsymbol{\theta})\right].$$

While the original likelihood contains the product of probability densities, the log-likelihood contains the sum of log probability densities, which is often easier to compute in practice. The likelihood equation for the log-likelihood is given by

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log L(\boldsymbol{\theta})\bigg|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}_{\mathrm{ML}}} = \mathbf{0}_b.$$

Below, MLE for the Gaussian model is explained in detail. MLE for the Gaussian mixture model will be explained in Chapter 15.

12.2 GAUSSIAN MODEL

In Section 4.2 and Section 6.2, the Gaussian distribution was introduced. The *Gaussian model* corresponds to a parametric model for the Gaussian distribution and is given for d-dimensional pattern $\boldsymbol{x}$ as

$$q(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{d}{2}}\det(\boldsymbol{\Sigma})^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^{\top}\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right).$$

Here, d-dimensional vector $\boldsymbol{\mu}$ and $d \times d$ matrix $\boldsymbol{\Sigma}$ are the parameters of the Gaussian model, and $\det(\cdot)$ denotes the determinant. $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ correspond to the expectation

The following formulas hold for vector and matrix derivatives:

$$\frac{\partial \boldsymbol{\mu}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}{\partial \boldsymbol{\mu}} = 2\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}, \quad \frac{\partial \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu}}{\partial \boldsymbol{\mu}} = \boldsymbol{\Sigma}^{-1}\boldsymbol{x},$$

$$\frac{\partial \boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{x}}{\partial \boldsymbol{\Sigma}} = -\boldsymbol{\Sigma}^{-1}\boldsymbol{x}\boldsymbol{x}^\top \boldsymbol{\Sigma}^{-1}, \quad \frac{\partial \log(\det(\boldsymbol{\Sigma}))}{\partial \boldsymbol{\Sigma}} = \boldsymbol{\Sigma}^{-1}, \quad \frac{\partial \mathrm{tr}(\widetilde{\boldsymbol{\Sigma}}^{-1}\boldsymbol{\Sigma})}{\partial \boldsymbol{\Sigma}} = \widetilde{\boldsymbol{\Sigma}}^{-1}.$$

FIGURE 12.3

Formulas for vector and matrix derivatives [80].

matrix and the variance-covariance matrix, respectively,

$$\boldsymbol{\mu} = E[\boldsymbol{x}] = \int \boldsymbol{x} q(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \mathrm{d}\boldsymbol{x},$$

$$\boldsymbol{\Sigma} = V[\boldsymbol{x}] = \int (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^\top q(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) \mathrm{d}\boldsymbol{x}.$$

For i.i.d. training samples $\{\boldsymbol{x}_i\}_{i=1}^n$ following the Gaussian model $q(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$, the log-likelihood is given by

$$\log L(\boldsymbol{\mu}, \boldsymbol{\Sigma}) = -\frac{nd \log 2\pi}{2} - \frac{n \log(\det(\boldsymbol{\Sigma}))}{2} - \frac{1}{2}\sum_{i=1}^n (\boldsymbol{x}_i - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{\mu}).$$

The likelihood equation for the Gaussian model is given as

$$\begin{cases} \dfrac{\partial}{\partial \boldsymbol{\mu}} \log L(\boldsymbol{\mu}, \boldsymbol{\Sigma})\Big|_{\boldsymbol{\mu}=\widehat{\boldsymbol{\mu}}_{\mathrm{ML}}} = \boldsymbol{0}_d, \\ \dfrac{\partial}{\partial \boldsymbol{\Sigma}} \log L(\boldsymbol{\mu}, \boldsymbol{\Sigma})\Big|_{\boldsymbol{\Sigma}=\widehat{\boldsymbol{\Sigma}}_{\mathrm{ML}}} = \boldsymbol{O}_{d\times d}, \end{cases}$$

where $\boldsymbol{O}_{d\times d}$ denotes the $d \times d$ zero matrix.

For deriving the maximum likelihood solutions, formulas for vector and matrix derivatives shown in Fig. 12.3 are useful. Indeed, the partial derivatives of the log-likelihood with respect to vector $\boldsymbol{\mu}$ and matrix $\boldsymbol{\Sigma}$ are given by

$$\frac{\partial \log L}{\partial \boldsymbol{\mu}} = n\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu} + \boldsymbol{\Sigma}^{-1}\sum_{i=1}^n \boldsymbol{x}_i,$$

$$\frac{\partial \log L}{\partial \boldsymbol{\Sigma}} = -\frac{n}{2}\boldsymbol{\Sigma}^{-1} + \frac{1}{2}\boldsymbol{\Sigma}^{-1}\left(\sum_{i=1}^n (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top\right)\boldsymbol{\Sigma}^{-1}.$$

Then the maximum likelihood estimators $\widehat{\boldsymbol{\mu}}_{\mathrm{ML}}$ and $\widehat{\boldsymbol{\Sigma}}_{\mathrm{ML}}$ are given as

$$\widehat{\boldsymbol{\mu}}_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i,$$

$$\widehat{\boldsymbol{\Sigma}}_{\mathrm{ML}} = \frac{1}{n}\sum_{i=1}^{n} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_{\mathrm{ML}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_{\mathrm{ML}})^\top,$$

which correspond to the *sample mean* and the *sample variance-covariance matrix*. Here, we assumed that we have enough training samples so that $\widehat{\boldsymbol{\Sigma}}_{\mathrm{ML}}$ is invertible.

The above Gaussian model considered a generic variance-covariance matrix $\boldsymbol{\Sigma}$, but a slightly simplified one having no correlation can also be considered (see Fig. 6.1). This corresponds to restricting $\boldsymbol{\Sigma}$ to be a diagonal matrix:

$$\boldsymbol{\Sigma} = \mathrm{diag}\left((\sigma^{(1)})^2, \ldots, (\sigma^{(d)})^2\right).$$

Then the Gaussian model having no correlation can be expressed as

$$q(\boldsymbol{x}; \boldsymbol{\mu}, \sigma^{(1)}, \ldots, \sigma^{(d)}) = \prod_{j=1}^{d} \frac{1}{\sqrt{2\pi(\sigma^{(j)})^2}} \exp\left(-\frac{(x^{(j)} - \mu^{(j)})^2}{2(\sigma^{(j)})^2}\right),$$

where $x^{(j)}$ and $\mu^{(j)}$ denotes the jth elements of d-dimensional vectors $\boldsymbol{x}$ and $\boldsymbol{\mu}$, respectively. The maximum likelihood solution for $\sigma^{(j)}$ is given by

$$\widehat{\sigma}^{(j)}_{\mathrm{ML}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} (x_i^{(j)} - \mu_i^{(j)})^2}.$$

The Gaussian model can be further simplified if all variances $(\sigma^{(j)})^2$ are assumed to be equal. Denoting the common variance by σ^2, the Gaussian model is expressed as

$$q(\boldsymbol{x}; \boldsymbol{\mu}, \sigma) = \frac{1}{(2\pi\sigma^2)^{\frac{d}{2}}} \exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{\mu})^\top (\boldsymbol{x} - \boldsymbol{\mu})}{2\sigma^2}\right).$$

Then the maximum likelihood solution for σ is given by

$$\widehat{\sigma}_{\mathrm{ML}} = \sqrt{\frac{1}{nd}\sum_{i=1}^{n} (\boldsymbol{x}_i - \boldsymbol{\mu})^\top (\boldsymbol{x}_i - \boldsymbol{\mu})} = \sqrt{\frac{1}{d}\sum_{j=1}^{d} (\widehat{\sigma}^{(j)}_{\mathrm{ML}})^2}.$$

A MATLAB code for MLE with one-dimensional Gaussian model is given in Fig. 12.4, and its behavior is illustrated in Fig. 12.5.

12.3 COMPUTING THE CLASS-POSTERIOR PROBABILITY

Let us come back to the pattern recognition problem and learn the class-conditional probability density $p(\boldsymbol{x}|y)$ by MLE with Gaussian models:

```
n=5; m=0; s=1; x=s*randn(n,1)+m; mh=mean(x); sh=std(x,1);
X=linspace(-4,4,100); Y=exp(-(X-m).^2./(2*s^2))/(2*pi*s);
Yh=exp(-(X-mh).^2./(2*sh^2))/(2*pi*sh);

figure(1); clf; hold on;
plot(X,Y,'r-',X,Yh,'b--',x,zeros(size(x)),'ko');
legend('True','Estimated');
```

FIGURE 12.4

MATLAB code for MLE with one-dimensional Gaussian model.

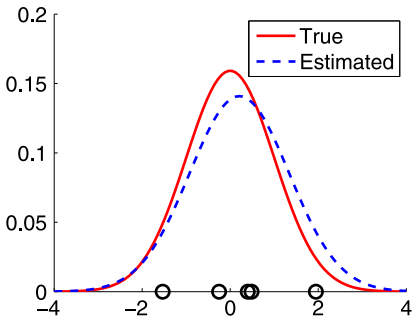


FIGURE 12.5

Example of MLE with one-dimensional Gaussian model.

$$\widehat{p}(\boldsymbol{x}|y) = \frac{1}{(2\pi)^{\frac{d}{2}} \det(\widehat{\boldsymbol{\Sigma}}_y)^{\frac{1}{2}}} \exp\left(-\frac{1}{2}(\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_y)^\top \widehat{\boldsymbol{\Sigma}}_y^{-1} (\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_y)\right).$$

Here, $\widehat{\boldsymbol{\mu}}_y$ and $\widehat{\boldsymbol{\Sigma}}_y$ are the maximum likelihood estimators of the expectation and variance-covariance matrices for class y:

$$\widehat{\boldsymbol{\mu}}_y = \frac{1}{n_y} \sum_{i:y_i=y} \boldsymbol{x}_i, \tag{12.2}$$

$$\widehat{\boldsymbol{\Sigma}}_y = \frac{1}{n_y} \sum_{i:y_i=y} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_y)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_y)^\top, \tag{12.3}$$

where n_y denotes the number of training samples in class y.

As shown in Fig. 12.2, the log function is monotone increasing and maximizing the log class-posterior probability is often more convenient than maximizing the plain

class-posterior probability. From the Bayes' theorem explained in Section 5.4,

$$p(y|\boldsymbol{x}) = \frac{p(\boldsymbol{x}|y)p(y)}{p(\boldsymbol{x})},$$

the log class-posterior probability is expressed as

$$\log p(y|\boldsymbol{x}) = \log p(\boldsymbol{x}|y) + \log p(y) - \log p(\boldsymbol{x}).$$

As shown in Eq. (11.4), the class-prior probability is simply estimated by the ratio of training samples:

$$\widehat{p}(y) = \frac{n_y}{n}.$$

Then the log class-posterior probability $\log p(y|\boldsymbol{x})$ can be estimated as

$$\begin{aligned}
\log \widehat{p}(y|\boldsymbol{x}) &= \log \widehat{p}(\boldsymbol{x}|y) + \log \widehat{p}(y) - \log p(\boldsymbol{x}) \\
&= -\frac{d}{2}\log(2\pi) - \frac{1}{2}\log(\det(\widehat{\boldsymbol{\Sigma}}_y)) - \frac{1}{2}(\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_y)^\top \widehat{\boldsymbol{\Sigma}}_y^{-1}(\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_y) \\
&\quad + \log \frac{n_y}{n} - \log p(\boldsymbol{x}) \\
&= -\frac{1}{2}(\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_y)^\top \widehat{\boldsymbol{\Sigma}}_y^{-1}(\boldsymbol{x} - \widehat{\boldsymbol{\mu}}_y) - \frac{1}{2}\log(\det(\widehat{\boldsymbol{\Sigma}}_y)) + \log \frac{n_y}{n} + C,
\end{aligned}$$

where C is a constant independent of y. As shown above, if $p(\boldsymbol{x}|y)$ is estimated by a Gaussian model, $\log \widehat{p}(y|\boldsymbol{x})$ becomes a quadratic function of $\boldsymbol{x}$.

The first term in the above equation,

$$(\boldsymbol{x} - \widehat{\boldsymbol{\mu}})^\top \widehat{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{x} - \widehat{\boldsymbol{\mu}}),$$

is called the *Mahalanobis distance* between $\boldsymbol{x}$ and $\widehat{\boldsymbol{\mu}}$. The Mahalanobis distance regards the set of points on a *hyperellipsoid* specified by $\widehat{\boldsymbol{\Sigma}}$ as the same distance. To understand this, let us eigendecompose (see Fig. 6.2) the variance-covariance matrix $\widehat{\boldsymbol{\Sigma}}$ as

$$\widehat{\boldsymbol{\Sigma}}\boldsymbol{\phi} = \lambda\boldsymbol{\phi},$$

where $\{\lambda_j\}_{j=1}^d$ are the eigenvalues of $\boldsymbol{\Sigma}$ and $\{\boldsymbol{\phi}_j\}_{j=1}^d$ are the corresponding eigenvectors that are normalized to have a unit norm. Then $\widehat{\boldsymbol{\Sigma}}$ can be expressed as

$$\widehat{\boldsymbol{\Sigma}} = \sum_{j=1}^{d} \lambda_j \boldsymbol{\phi}_j \boldsymbol{\phi}_j^\top,$$

and $\widehat{\boldsymbol{\Sigma}}^{-1}$ is written explicitly as

$$\widehat{\boldsymbol{\Sigma}}^{-1} = \sum_{j=1}^{d} \frac{1}{\lambda_j} \boldsymbol{\phi}_j \boldsymbol{\phi}_j^\top.$$

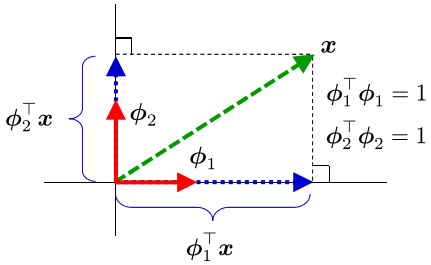


FIGURE 12.6

Orthogonal projection.

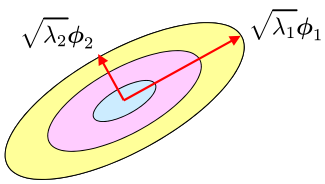


FIGURE 12.7

Mahalanobis distance having hyperellipsoidal contours.

Since $\{\boldsymbol{\phi}_j\}_{j=1}^d$ are mutually orthogonal, the Mahalanobis distance can be expressed as

$$(\boldsymbol{x} - \widehat{\boldsymbol{\mu}})^\top \widehat{\boldsymbol{\Sigma}}^{-1} (\boldsymbol{x} - \widehat{\boldsymbol{\mu}}) = \sum_{j=1}^{d} \frac{(\boldsymbol{\phi}_j^\top (\boldsymbol{x} - \widehat{\boldsymbol{\mu}}))^2}{\lambda_j},$$

$\boldsymbol{\phi}_j^\top (\boldsymbol{x} - \widehat{\boldsymbol{\mu}})$ corresponds to the length of the projection of $\boldsymbol{x} - \widehat{\boldsymbol{\mu}}$ onto $\boldsymbol{\phi}_j$ (see Fig. 12.6). Thus, the Mahalanobis distance is the squared sum of the projection lengths divided by λ_j, and the set of points having the same Mahalanobis distance (say, 1) satisfies

$$\sum_{j=1}^{d} \frac{1}{\lambda_j} (\boldsymbol{\phi}_j^\top (\boldsymbol{x} - \widehat{\boldsymbol{\mu}}))^2 = 1.$$

This represents the hyperellipsoid with the *principal axes* parallel to $\{\boldsymbol{\phi}_j\}_{j=1}^d$ and length $\{\sqrt{\lambda_j}\}_{j=1}^d$. When the dimensionality of $\boldsymbol{x}$ is $d = 2$, the hyperellipsoid is reduced to the ellipse (Fig. 12.7).

When the number of classes is $c = 2$, the decision boundary is the set of points having the same class-posterior probability (i.e., 1/2):

$$p(y = 1|\boldsymbol{x}) = p(y = 2|\boldsymbol{x}). \tag{12.4}$$

Therefore, if $p(\boldsymbol{x}|y)$ is estimated by a Gaussian model, the decision boundary is a *quadratic hypersurface*.

12.4 FISHER'S LINEAR DISCRIMINANT ANALYSIS (FDA)

Suppose further that the variance-covariance matrix of $\boldsymbol{\Sigma}_y$ of each class is the same:

$$\boldsymbol{\Sigma}_1 = \cdots = \boldsymbol{\Sigma}_c = \boldsymbol{\Sigma},$$

where $\boldsymbol{\Sigma}$ is the common variance-covariance matrix that is independent of class y. Then the maximum likelihood solution $\widehat{\boldsymbol{\Sigma}}$ for the common variance-covariance matrix $\boldsymbol{\Sigma}$ is given by

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{y=1}^{c}\sum_{i:y_i=y}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_y)^\top(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_y) = \sum_{y=1}^{c}\frac{n_y}{n}\widehat{\boldsymbol{\Sigma}}_y. \tag{12.5}$$

This shows that $\widehat{\boldsymbol{\Sigma}}$ is the weighted average of each solution $\widehat{\boldsymbol{\Sigma}}_y$ (see Eq. (12.3)) according to the ratio of samples n_y/n.

With $\widehat{\boldsymbol{\Sigma}}$, the log class-posterior probability $\log p(y|\boldsymbol{x})$ can be estimated as

$$\begin{aligned}\log \widehat{p}(y|\boldsymbol{x}) &= -\frac{1}{2}\boldsymbol{x}^\top\widehat{\boldsymbol{\Sigma}}^{-1}\boldsymbol{x} + \boldsymbol{x}^\top\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}_y - \frac{1}{2}\widehat{\boldsymbol{\mu}}_y^\top\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}_y \\ &\quad - \frac{1}{2}\log(\det(\widehat{\boldsymbol{\Sigma}})) + \log\frac{n_y}{n} + C \\ &= \boldsymbol{x}^\top\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}_y - \frac{1}{2}\widehat{\boldsymbol{\mu}}_y^\top\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}_y + \log\frac{n_y}{n} + C',\end{aligned} \tag{12.6}$$

where C' is a constant independent of y. Thus, when the variance-covariance matrix is common to all classes, the log class-posterior probability is a linear function of $\boldsymbol{x}$.

Eq. (12.4) implies that the decision boundary when the number of classes is $c = 2$ is given by

$$\widehat{\boldsymbol{a}}^\top\boldsymbol{x} + \widehat{b} = 0,$$

where

$$\begin{aligned}\widehat{\boldsymbol{a}} &= \widehat{\boldsymbol{\Sigma}}^{-1}(\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_2), \\ \widehat{b} &= -\frac{1}{2}(\widehat{\boldsymbol{\mu}}_1^\top\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}_1 - \widehat{\boldsymbol{\mu}}_2^\top\widehat{\boldsymbol{\Sigma}}^{-1}\widehat{\boldsymbol{\mu}}_2) + \log\frac{n_1}{n_2}.\end{aligned}$$

This shows that the decision boundary is a *hyperplane*. This classification method is called *Fisher's linear discriminant analysis* (FDA) [41].

Let us compute the FDA solution for two-dimensional training samples plotted in Fig. 12.8, where "$\circ$" and "$\times$" denote training samples in class 1 and class 2 for $n = 300$ and $n_1 = n_2 = 150$. The Gaussian models with common variance-covariance matrix are used to approximate $p(\boldsymbol{x}|y = 1)$ and $p(\boldsymbol{x}|y = 2)$, and the following maximum likelihood estimators are obtained:

$$\widehat{\boldsymbol{\mu}}_1 = \begin{pmatrix}2\\0\end{pmatrix}, \quad \widehat{\boldsymbol{\mu}}_2 = \begin{pmatrix}-2\\0\end{pmatrix}, \quad \text{and} \quad \widehat{\boldsymbol{\Sigma}} = \begin{pmatrix}1 & 0\\0 & 9\end{pmatrix}.$$

Then the FDA solution is given by

$$\widehat{\boldsymbol{a}} = \begin{pmatrix}4\\0\end{pmatrix} \quad \text{and} \quad \widehat{b} = 0.$$

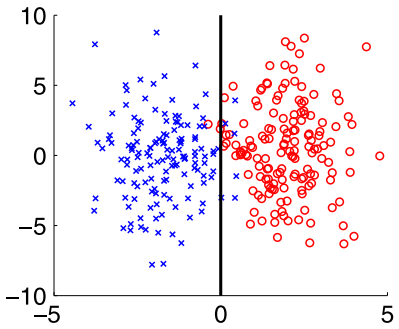
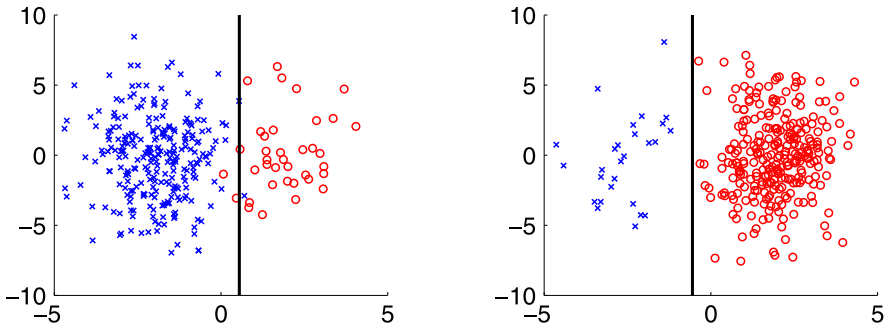


FIGURE 12.8

Linear discriminant analysis.



(a) $n_1/n_2 = 1/9$ (b) $n_1/n_2 = 9$

FIGURE 12.9

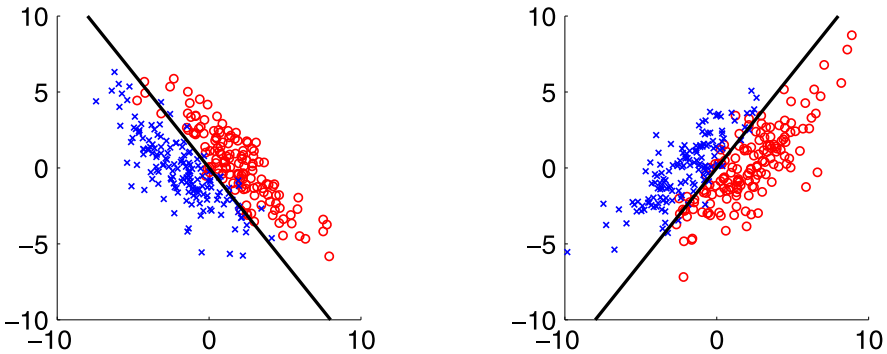
When the classwise sample ratio n_1/n_2 is changed.

The obtained decision boundary is plotted in Fig. 12.8, showing that the FDA solution separates the two Gaussian distributions in the middle.

If the classwise sample ratio n_1/n_2 is changed, the decision boundary yields

$$\widehat{\boldsymbol{a}} = \begin{pmatrix} 4 \\ 0 \end{pmatrix} \quad \text{and} \quad \widehat{b} = \log \frac{n_1}{n_2}.$$

As illustrated in Fig. 12.9, the decision boundary shifts depending on the classwise sample ratio n_1/n_2.



(a) $\beta = \pi/4$ (b) $\beta = -\pi/4$

FIGURE 12.10

When the classwise sample distributions are rotated.

Next, let us set $n_1/n_2 = 1$ again and change the common variance-covariance matrix $\widehat{\boldsymbol{\Sigma}}$ to

$$\widehat{\boldsymbol{\Sigma}} = \begin{pmatrix} 9 - 8\cos^2\beta & 8\sin\beta\cos\beta \\ 8\sin\beta\cos\beta & 9 - 8\sin^2\beta \end{pmatrix},$$

which corresponds to rotating the classwise sample distributions by angle β. Then the FDA solution is given by

$$\widehat{\boldsymbol{a}} = \begin{pmatrix} 1 + 8\cos^2\beta \\ -8\sin\beta\cos\beta \end{pmatrix} \quad \text{and} \quad \widehat{b} = 0,$$

where

$$\widehat{\boldsymbol{\Sigma}}^{-1} = \begin{pmatrix} 1 - \frac{8}{9}\sin^2\beta & -\frac{8}{9}\sin\beta\cos\beta \\ -\frac{8}{9}\sin\beta\cos\beta & 1 - \frac{8}{9}\cos^2\beta \end{pmatrix}$$

is used in the derivation. As illustrated in Fig. 12.10, the decision boundary rotates according to the rotation of the data set.

12.5 HAND-WRITTEN DIGIT RECOGNITION

In this section, FDA is applied to hand-written digit recognition using MATLAB. The digit data set available from

`http://www.ms.k.u-tokyo.ac.jp/sugi/data/digit.mat`

is used.

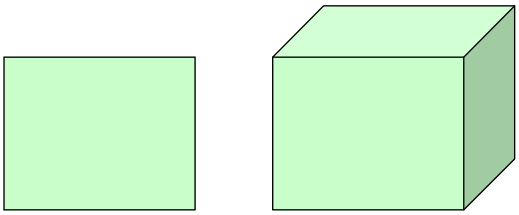


FIGURE 12.11

Matrix and third tensor.

12.5.1 PREPARATION

First, let us load the digit data in the memory space as follows.

```
> load digit.mat
```

The `whos` function shows that variable `X` contains digit data for training and variable `T` contains digit data for testing:

```
> whos

  Name        Size                Bytes  Class

  T         256x200x10          4096000  double
  X         256x500x10         10240000  double
```

Here, `X` and `T` are the third *tensors*, meaning that they are arrays taking three arguments (see Fig. 12.11). Each digit image is represented by a 256-dimensional vector, which is a vectorized expression of a 16×16 pixel image, as explained in Fig. 11.1(b). Each element take a real value in $[-1, 1]$, where -1 corresponds to black and 1 corresponds to white. `X` contains 5000 digit samples (500 samples for each digit from "0" to "9") and `T` contains 2000 digit samples (200 samples for each digit from "0" to "9").

For example, the 23rd training sample of digit "5" can be extracted into variable `x` as follows:

```
> x=X(:,23,5);
```

Note that digit "0" corresponds to 10 in the third argument. The extracted sample can be visualized as follows:

```
> imagesc(reshape(x,[16 16])')
```

12.5.2 IMPLEMENTING LINEAR DISCRIMINANT ANALYSIS

Let us implement FDA for classifying digits "1" and "2." First, let us clear the memory space and reload the digit data:

```
> clear all
> load digit.mat
```

Suppose that patterns of digits "1" and "2" follow the normal distribution with a common variance-covariance matrix. The sample means of each class can be obtained as follows (see Eq. (12.2)):

```
> mu1=mean(X(:,:,1),2);
> mu2=mean(X(:,:,2),2);
```

The common variance-covariance matrix can be obtained as follows (see Eq. (12.5)):

```
> S=(cov(X(:,:,1)')+cov(X(:,:,2)'))/2;
```

Then the class-posterior probability for a test pattern can be computed as follows (see Eq. (12.6)):

```
> t=T(:,1,2);
> invS=inv(S+0.000001*eye(256));
> p1=mu1'*invS*t-mu1'*invS*mu1/2;
> p2=mu2'*invS*t-mu2'*invS*mu2/2;
```

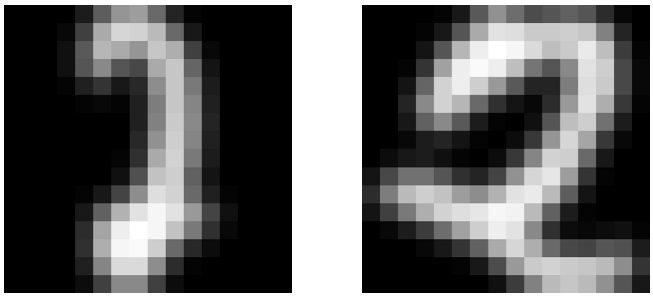
Note that irrelevant constants are ignored here, and `0.000001*eye(256)` is added to `S` to avoid numerical problems when it is inverted, where `eye(256)` denotes the 256×256 identity matrix. Investigating the difference of the class-posterior probabilities allows us to classify the test pattern:

```
> sign(p1-p2)
ans = -1
```

In this example, the test pattern is classified into class "2," meaning that the classification is successful. If `sign(p1-p2)` is 1, the test pattern is classified into class "1."

Classification results for all test patterns in class "2" can be obtained as follows:

```
> t=T(:,:,2);
> p1=mu1'*invS*t-mu1'*invS*mu1/2;
> p2=mu2'*invS*t-mu2'*invS*mu2/2;
> result=sign(p1-p2);
```



(a) 69th test pattern (b) 180th test pattern

FIGURE 12.12

Misclassified test patterns.

Note that `p1` and `p2` are the horizontal vectors. The classification results can be summarized as follows:

```
> sum(result==-1)
ans = 198
> sum(result~=-1)
ans = 2
```

This shows that, among 200 test patterns in class "2," 198 samples are correctly classified and 2 samples are misclassified. Thus, the correct classification rate is 2/200 = 99%. Misclassified samples can be found as follows:

```
> find(result~=-1)
ans =
   69  180
```

These test patterns are actually difficult to classify even for human (in particular, the 69th pattern), as illustrated in Fig. 12.12:

```
> imagesc(reshape(t(:,69),[16 16])')
> imagesc(reshape(t(:,180),[16 16])')
```

12.5.3 MULTICLASS CLASSIFICATION

Next, digit samples in classes "1," "2," and "3" are classified by FDA, under the assumption that the variance-covariance matrix is common to all classes. A MATLAB code is provided in Fig. 12.13. Here the classification results are summarized in the *confusion matrix*, which contains the number of times patterns in class y is categorized into class y' in the (y, y') element.

```
clear all
load digit.mat X T
[d,m,c]=size(T); c=3;

S=zeros(d,d);
for y=1:c
  mu(:,y)=mean(X(:,:,y),2);
  S=S+cov(X(:,:,y)')/c;
end
h=inv(S)*mu;
for y=1:c
  p(:,:,y)=h'*T(:,:,y)-repmat(sum(mu.*h)',[1,m])/2;
end
[pmax P]=max(p);
P=squeeze(P);
for y=1:c
  C(:,y)=sum(P==y);
end
C
```

FIGURE 12.13

MATLAB code for multiclass classification by FDA.

The following confusion matrix is obtained:

```
> C
C =
  199    1    0
    0  192    8
    0    2  198
```

This means that

- 199 test patterns in class "1" are correctly classified and the remaining 1 pattern is misclassified as "2."
- 192 test patterns in class "2" are correctly classified and the remaining 8 patterns are misclassified as "3."
- 198 test patterns in class "3" are correctly classified and the remaining 3 patterns are misclassified as "2."

If `c=3;` is commented out in the program in Fig. 12.13, all patterns in all 10 classes are classified. The obtained confusion matrix is shown in Fig. 12.14, showing

Predicted class

True class	1	2	3	4	5	6	7	8	9	0
1	199	0	0	0	1	0	0	0	0	0
2	0	169	8	8	1	2	4	8	0	0
3	0	0	182	1	5	0	2	8	1	1
4	2	2	0	182	0	1	0	3	10	0
5	0	0	21	4	162	1	0	4	4	4
6	1	2	0	1	5	185	0	3	0	3
7	2	0	1	5	1	0	181	0	9	1
8	0	1	16	6	6	0	1	164	3	3
9	1	0	0	8	0	0	7	2	182	0
0	0	0	3	0	0	4	0	1	0	192

FIGURE 12.14

Confusion matrix for 10-class classification by FDA. The correct classification rate is 1798/2000 = 89.9%.

that 1798 test samples out of 2000 samples are correctly classified. Thus, the correct classification rate is

$$1798/2000 = 89.9\%.$$