CHAPTER

MAXIMUM LIKELIHOOD ESTIMATION FOR GAUSSIAN MIXTURE MODEL 15

CHAPTER CONTENTS

Fisher's linear discriminant analysis introduced in Chapter 12 is a simple and practical classification method. However, approximating class-conditional probability densities by the Gaussian models can be too restrictive in practice. In this chapter, a more expressive model called the *Gaussian mixture model* is introduced and its MLE is discussed.

15.1 GAUSSIAN MIXTURE MODEL

If patterns in a class are distributed in several clusters, approximating the class-conditional distribution by a single Gaussian model may not be appropriate. For example, Fig. 15.1(a) illustrates the situation where a *unimodal* distribution is approximated by a single Gaussian model, which results in accurate estimation. On the other hand, Fig. 15.1(b) illustrates the situation where a *multimodal* distribution is approximated by a single Gaussian model, which performs poorly even with a large number of training samples.

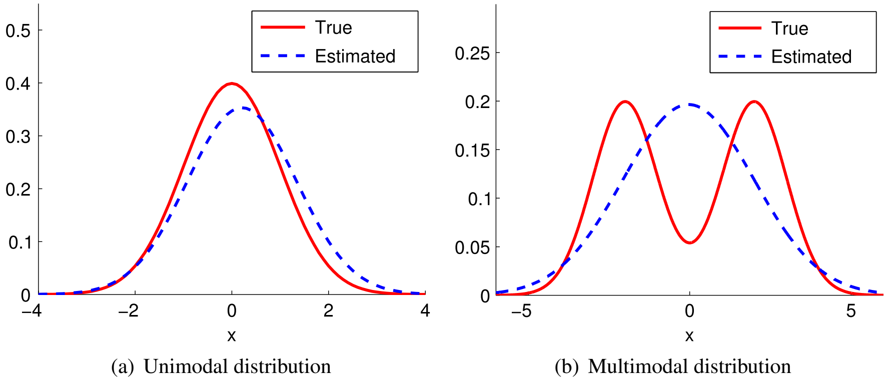
A *Gaussian mixture model*, defined by

$$q(\boldsymbol{x};\boldsymbol{\theta}) = \sum_{\ell=1}^{m} w_\ell N(\boldsymbol{x};\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell),$$

is suitable to approximate such multimodal distributions. Here, $N(\boldsymbol{x};\boldsymbol{\mu},\boldsymbol{\Sigma})$ denotes a Gaussian model with expectation $\boldsymbol{\mu}$ and variance-covariance matrix $\boldsymbol{\Sigma}$:

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

An Introduction to Statistical Machine Learning. DOI: 10.1016/B978-0-12-802121-7.00026-1



(a) Unimodal distribution (b) Multimodal distribution

FIGURE 15.1

MLE for Gaussian model.

Thus, a Gaussian mixture model is a *linear combination* of m Gaussian models weighted according to $\{w_\ell\}_{\ell=1}^m$. The parameter $\boldsymbol{\theta}$ of the Gaussian mixture model is given by

$$\boldsymbol{\theta} = (w_1, \ldots, w_m, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_m).$$

The Gaussian mixture model $q(\boldsymbol{x}; \boldsymbol{\theta})$ should satisfy the following condition to be a probability density function:

$$\forall \boldsymbol{x} \in \mathcal{X}, \;\; q(\boldsymbol{x}; \boldsymbol{\theta}) \geq 0 \;\;\text{and}\;\; \int_{\mathcal{X}} q(\boldsymbol{x}; \boldsymbol{\theta}) \mathrm{d}\boldsymbol{x} = 1.$$

To this end, $\{w_\ell\}_{\ell=1}^m$ are imposed to be

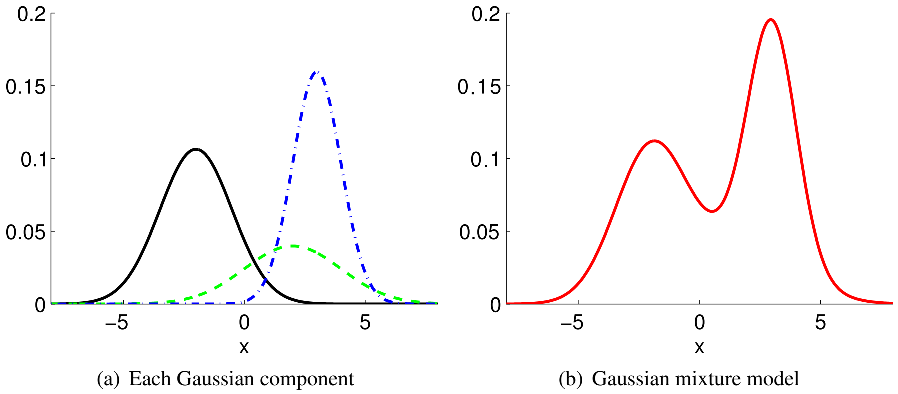
$$w_1, \ldots, w_m \geq 0 \;\;\text{and}\;\; \sum_{\ell=1}^m w_\ell = 1. \tag{15.1}$$

Fig. 15.2 illustrates an example of the Gaussian mixture model, which represents a multimodal distribution by linearly combining multiple Gaussian models.

15.2 MLE

The parameter $\boldsymbol{\theta}$ in the Gaussian mixture model is learned by MLE explained in Chapter 12. The likelihood is given by

$$L(\boldsymbol{\theta}) = \prod_{i=1}^n q(\boldsymbol{x}_i; \boldsymbol{\theta}), \tag{15.2}$$



(a) Each Gaussian component

(b) Gaussian mixture model

FIGURE 15.2

Example of Gaussian mixture model: $q(x) = 0.4N(x; -2, 1.5^2) + 0.2N(x; 2, 2^2) + 0.4N(x; 3, 1^2)$.

and MLE finds its maximizer with respect to $\boldsymbol{\theta}$. When the above likelihood is maximized for the Gaussian mixture model, the *constraints* given by Eq. (15.1) need to be satisfied:

$$\widehat{\boldsymbol{\theta}} = \operatorname*{argmax}_{\boldsymbol{\theta}} L(\boldsymbol{\theta})$$

$$\text{subject to} \quad w_1, \ldots, w_m \geq 0 \quad \text{and} \quad \sum_{\ell=1}^{m} w_\ell = 1.$$

Due to the constraints, the maximizer $\widehat{\boldsymbol{\theta}}$ cannot be simply obtained by setting the derivative of the likelihood to zero. Here, $w_1, \ldots, w_m$ are re-parameterized as

$$w_\ell = \frac{\exp(\gamma_\ell)}{\sum_{\ell'=1}^{m} \exp(\gamma_{\ell'})}, \tag{15.3}$$

and $\{\gamma_\ell\}_{\ell=1}^{m}$ are learned, which automatically fulfills Eq. (15.1).

The maximum likelihood solution $\widehat{\boldsymbol{\theta}}$ satisfies the following likelihood equation for $\log L(\boldsymbol{\theta})$:

$$\begin{cases} \left.\dfrac{\partial}{\partial \gamma_\ell} \log L(\boldsymbol{\theta})\right|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}} = 0, \\ \left.\dfrac{\partial}{\partial \boldsymbol{\mu}_\ell} \log L(\boldsymbol{\theta})\right|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}} = \boldsymbol{0}_d, \\ \left.\dfrac{\partial}{\partial \boldsymbol{\Sigma}_\ell} \log L(\boldsymbol{\theta})\right|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}} = \boldsymbol{O}_{d\times d}, \end{cases} \tag{15.4}$$

where $\mathbf{0}_d$ denotes the d-dimensional zero vector and $\boldsymbol{O}_{d\times d}$ denotes the $d \times d$ zero matrix. Substituting Eq. (15.3) into Eq. (15.2), the log-likelihood is expressed as

$$\log L(\boldsymbol{\theta}) = \sum_{i=1}^{n} \log \sum_{\ell=1}^{m} \exp(\gamma_\ell) N(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell) - n \log \sum_{\ell=1}^{m} \exp(\gamma_\ell).$$

Taking the partial derivative of the above log-likelihood with respect to γ_ℓ gives

$$\begin{aligned}\frac{\partial}{\partial \gamma_\ell} \log L(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \frac{\exp(\gamma_\ell) N(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)}{\sum_{\ell'=1}^{m} \exp(\gamma_{\ell'}) N(\boldsymbol{x}_i; \boldsymbol{\mu}_{\ell'}, \boldsymbol{\Sigma}_{\ell'})} - \frac{n\gamma_\ell}{\sum_{\ell'=1}^{m} \exp(\gamma_{\ell'})} \\ &= \sum_{i=1}^{n} \eta_{i,\ell} - n w_\ell,\end{aligned}$$

where $\eta_{i,\ell}$ is defined as

$$\eta_{i,\ell} = \frac{w_\ell N(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)}{\sum_{\ell'=1}^{m} w_{\ell'} N(\boldsymbol{x}_i; \boldsymbol{\mu}_{\ell'}, \boldsymbol{\Sigma}_{\ell'})}.$$

Similarly, taking the partial derivatives of the above log-likelihood with respect to $\boldsymbol{\mu}_\ell$ and $\boldsymbol{\Sigma}_\ell$ (see Fig. 12.3 for the derivative formulas) gives

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\mu}_\ell} \log L(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \eta_{i,\ell} \boldsymbol{\Sigma}_\ell^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_\ell), \\ \frac{\partial}{\partial \boldsymbol{\Sigma}_\ell} \log L(\boldsymbol{\theta}) &= \frac{1}{2} \sum_{i=1}^{n} \eta_{i,\ell} \left(\boldsymbol{\Sigma}_\ell^{-1} (\boldsymbol{x}_i - \boldsymbol{\mu}_\ell)(\boldsymbol{x}_i - \boldsymbol{\mu}_\ell)^\top \boldsymbol{\Sigma}_\ell^{-1} - \boldsymbol{\Sigma}_\ell^{-1} \right).\end{aligned}$$

Setting the above derivatives to zero shows that the maximum likelihood solution $\widehat{w}_\ell$, $\widehat{\boldsymbol{\mu}}_\ell$, and $\widehat{\boldsymbol{\Sigma}}_\ell$ should satisfy

$$\begin{cases} \widehat{w}_\ell = \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \widehat{\eta}_{i,\ell}, \\ \widehat{\boldsymbol{\mu}}_\ell = \dfrac{\sum_{i=1}^{n} \widehat{\eta}_{i,\ell} \boldsymbol{x}_i}{\sum_{i'=1}^{n} \widehat{\eta}_{i',\ell}}, \\ \widehat{\boldsymbol{\Sigma}}_\ell = \dfrac{\sum_{i=1}^{n} \widehat{\eta}_{i,\ell} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_\ell)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_\ell)^\top}{\sum_{i'=1}^{n} \widehat{\eta}_{i',\ell}}, \end{cases} \tag{15.5}$$

where $\widehat{\eta}_{i,\ell}$ is called the *responsibility* of the ℓth component for sample $\boldsymbol{x}_i$:

$$\widehat{\eta}_{i,\ell} = \frac{\widehat{w}_\ell N(\boldsymbol{x}_i; \widehat{\boldsymbol{\mu}}_\ell, \widehat{\boldsymbol{\Sigma}}_\ell)}{\sum_{\ell'=1}^{m} \widehat{w}_{\ell'} N(\boldsymbol{x}_i; \widehat{\boldsymbol{\mu}}_{\ell'}, \widehat{\boldsymbol{\Sigma}}_{\ell'})}. \tag{15.6}$$

In the above likelihood equation, variables are entangled in a complicated way and there is no known method to solve it analytically. Below, two *iterative algorithms* are introduced to numerically find a solution: the *gradient method* ad the *expectation-maximization (EM) algorithm*.

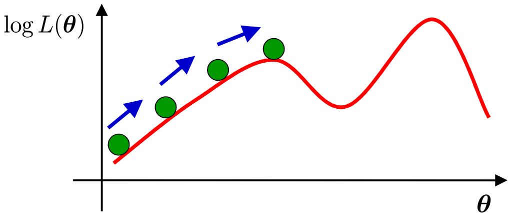


FIGURE 15.3

Schematic of gradient ascent.

1. Initialize the solution $\widehat{\boldsymbol{\theta}}$.
2. Compute the gradient of log-likelihood $\log L(\boldsymbol{\theta})$ at the current solution $\widehat{\boldsymbol{\theta}}$:

$$\frac{\partial}{\partial \boldsymbol{\theta}} \log L(\boldsymbol{\theta}) \bigg|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}}.$$

3. Update the parameter to go up the gradient:

$$\widehat{\boldsymbol{\theta}} \longleftarrow \widehat{\boldsymbol{\theta}} + \varepsilon \frac{\partial}{\partial \boldsymbol{\theta}} \log L(\boldsymbol{\theta}) \bigg|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}},$$

   where ε is a small positive scalar.
4. Iterate 2–3 until convergence.

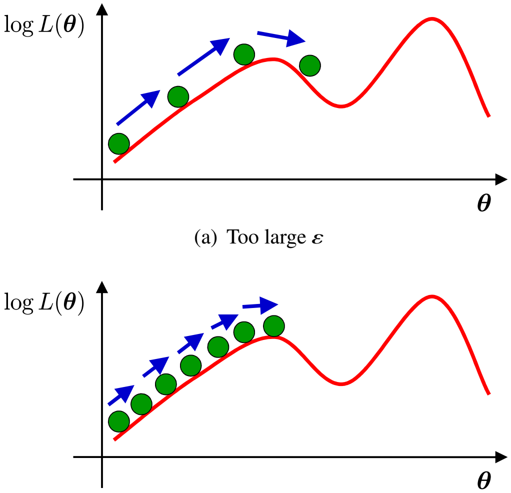
FIGURE 15.4

Algorithm of gradient ascent.

15.3 GRADIENT ASCENT ALGORITHM

A *gradient method* is a generic and simple optimization approach that iteratively updates the parameter to go up (down in the case of minimization) the gradient of an *objective function* (Fig. 15.3). The algorithm of gradient ascent is summarized in Fig. 15.4. Under a mild assumption, a gradient ascent solution is guaranteed to be *local optimal*, which corresponds to a peak of a local mountain and the objective value cannot be increased by any local parameter update.

A stochastic variant of the gradient method is to randomly choose a sample and update the parameter to go up the gradient for the selected sample. Such a stochastic method, called the *stochastic gradient* algorithm, was also shown to produce to a local optimal solution [4].



(a) Too large ε

(b) Too small ε

FIGURE 15.5

Step size ε in gradient ascent. The gradient flow can overshoot the peak if ε is large, while gradient ascent is slow if ε is too small.

Note that the (stochastic) gradient method does not only necessarily give the *global optimal solution* but also a local optimal solution, as illustrated in Fig. 15.3. Furthermore, its performance relies on the choice of the step size ε, which is not easy to determine in practice. If ε is large, gradient ascent is fast in the beginning, but the gradient flow can overshoot the peak (Fig. 15.5(a)). On the other hand, if ε is small, the peak may be found, but gradient ascent is slow in the beginning (Fig. 15.5(b)). To overcome this problem, starting from a large ε and then reducing ε gradually, called *simulated annealing*, would be useful. However, the choice of initial ε and the decreasing factor of ε is not straightforward in practice.

To mitigate the problem that only a local optimal solution can be found, it is practically useful to run the gradient algorithm multiple times from different initial solutions and choose the one that gives the best solution.

15.4 EM ALGORITHM

The difficulty of tuning the step size ε in the gradient method can be overcome by the EM algorithm [36]. The EM algorithm was originally developed for obtaining a maximum likelihood solution when input $\boldsymbol{x}$ is only partially observable. MLE for Gaussian mixture models can actually be regarded as learning from incomplete data, and the EM algorithm gives an efficient means to obtain a local optimal solution.

```
1. Initialize parameters $\{\widehat{w}_\ell, \widehat{\boldsymbol{\mu}}_\ell, \widehat{\boldsymbol{\Sigma}}_\ell\}_{\ell=1}^m$.
2. E-step: Compute responsibilities $\{\widehat{\eta}_{i,\ell}\}_{i=1,\ell=1}^{n\ \ m}$ from current parameters $\{\widehat{w}_\ell, \widehat{\boldsymbol{\mu}}_\ell, \widehat{\boldsymbol{\Sigma}}_\ell\}_{\ell=1}^m$:
       $$\widehat{\eta}_{i,\ell} \longleftarrow \frac{\widehat{w}_\ell N(\boldsymbol{x}_i; \widehat{\boldsymbol{\mu}}_\ell, \widehat{\boldsymbol{\Sigma}}_\ell)}{\sum_{\ell'=1}^m \widehat{w}_{\ell'} N(\boldsymbol{x}_i; \widehat{\boldsymbol{\mu}}_{\ell'}, \widehat{\boldsymbol{\Sigma}}_{\ell'})}.$$
3. M-step: Update parameters $\{\widehat{w}_\ell, \widehat{\boldsymbol{\mu}}_\ell, \widehat{\boldsymbol{\Sigma}}_\ell\}_{\ell=1}^m$ from current responsibilities $\{\widehat{\eta}_{i,\ell}\}_{i=1,\ell=1}^{n\ \ m}$:
       $$\widehat{w}_\ell \longleftarrow \frac{1}{n}\sum_{i=1}^n \widehat{\eta}_{i,\ell},$$
       $$\widehat{\boldsymbol{\mu}}_\ell \longleftarrow \frac{\sum_{i=1}^n \widehat{\eta}_{i,\ell}\boldsymbol{x}_i}{\sum_{i'=1}^n \widehat{\eta}_{i',\ell}},$$
       $$\widehat{\boldsymbol{\Sigma}}_\ell \longleftarrow \frac{\sum_{i=1}^n \widehat{\eta}_{i,\ell}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_\ell)(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}_\ell)^\top}{\sum_{i'=1}^n \widehat{\eta}_{i',\ell}},$$
4. Iterate 2–3 until convergence.
```

FIGURE 15.6

EM algorithm.

As summarized in Fig. 15.6, the EM algorithm consists of the E-step and the M-step, which correspond to updating the solution based on necessary condition (15.5) and computing its auxiliary variable (15.6) alternately.

The E-step and the M-step can be interpreted as follows:

E-step: A lower bound $b(\boldsymbol{\theta})$ of log-likelihood $\log L(\boldsymbol{\theta})$ that touches at current solution $\widehat{\boldsymbol{\theta}}$ is obtained:

$$\forall\boldsymbol{\theta}, \quad \log L(\boldsymbol{\theta}) \geq b(\boldsymbol{\theta}), \quad \text{and} \quad \log L(\widehat{\boldsymbol{\theta}}) = b(\widehat{\boldsymbol{\theta}}).$$

Note that this lower-bounding step corresponds to computing the expectation over unobserved variables, which is why this step is called the E-step.

M-step: The maximizer $\widehat{\boldsymbol{\theta}}'$ of the lower bound $b(\boldsymbol{\theta})$ is obtained.

$$\widehat{\boldsymbol{\theta}}' = \underset{\boldsymbol{\theta}}{\operatorname{argmax}}\, b(\boldsymbol{\theta}).$$

As illustrated in Fig. 15.7, iterating the E-step and the M-step increases the log-likelihood (precisely, the log-likelihood is monotone nondecreasing).

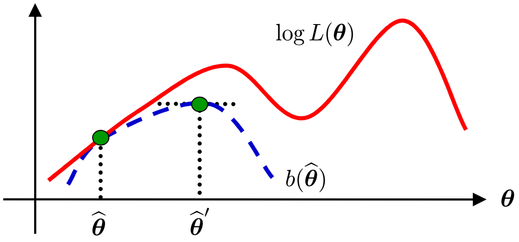


FIGURE 15.7

Maximizing the lower bound $b(\boldsymbol{\theta})$ of the log-likelihood $\log L(\boldsymbol{\theta})$.

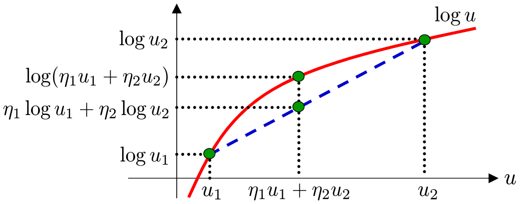


FIGURE 15.8

Jensen's inequality for $m = 2$. log is a concave function.

The lower bound in the E-step is derived based on *Jensen's inequality* explained in Section 8.3.1: for $\eta_1, \ldots, \eta_m \geq 0$ and $\sum_{\ell=1}^{m} \eta_\ell = 1$,

$$\log\left(\sum_{\ell=1}^{m} \eta_\ell u_\ell\right) \geq \sum_{\ell=1}^{m} \eta_\ell \log u_\ell. \tag{15.7}$$

For $m = 2$, Jensen's inequality is simplified as

$$\log\left(\eta_1 u_1 + \eta_2 u_2\right) \geq \eta_1 \log u_1 + \eta_2 \log u_2, \tag{15.8}$$

which can be intuitively understood by the *concavity* of the log function (see Fig. 15.8).

The log-likelihood $\log L(\boldsymbol{\theta})$ can be expressed by using the responsibility $\widehat{\eta}_{i,\ell}$ (see Eq. (15.6)) as

$$\begin{aligned}\log L(\boldsymbol{\theta}) &= \sum_{i=1}^{n} \log\left(\sum_{\ell=1}^{m} w_\ell N(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)\right)\\ &= \sum_{i=1}^{n} \log\left(\sum_{\ell=1}^{m} \widehat{\eta}_{i,\ell} \frac{w_\ell N(\boldsymbol{x}_i; \boldsymbol{\mu}_\ell, \boldsymbol{\Sigma}_\ell)}{\widehat{\eta}_{i,\ell}}\right).\end{aligned} \tag{15.9}$$

By associating $w_\ell N(\boldsymbol{x}_i;\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)/\widehat{\eta}_{i,\ell}$ in Eq. (15.9) with u_ℓ in Jensen's inequality (15.7), lower bound $b(\boldsymbol{\theta})$ of the log-likelihood $\log L(\boldsymbol{\theta})$ can be obtained as

$$\log L(\boldsymbol{\theta}) \geq \sum_{i=1}^{n}\sum_{\ell=1}^{m}\widehat{\eta}_{i,\ell}\log\left(\frac{w_\ell N(\boldsymbol{x}_i;\boldsymbol{\mu}_\ell,\boldsymbol{\Sigma}_\ell)}{\widehat{\eta}_{i,\ell}}\right) = b(\boldsymbol{\theta}).$$

This lower bound $b(\boldsymbol{\theta})$ touches $\log L(\boldsymbol{\theta})$ when $\boldsymbol{\theta} = \widehat{\boldsymbol{\theta}}$, because Eq. (15.6) implies

$$\begin{aligned} b(\widehat{\boldsymbol{\theta}}) &= \sum_{i=1}^{n}\left(\sum_{\ell=1}^{m}\widehat{\eta}_{i,\ell}\right)\log\left(\frac{\widehat{w}_\ell N(\boldsymbol{x}_i;\widehat{\boldsymbol{\mu}}_\ell,\widehat{\boldsymbol{\Sigma}}_\ell)}{\widehat{\eta}_{i,\ell}}\right)\\ &= \sum_{i=1}^{n}\log\left(\sum_{\ell'=1}^{m}\widehat{w}_{\ell'} N(\boldsymbol{x}_i;\widehat{\boldsymbol{\mu}}_{\ell'},\widehat{\boldsymbol{\Sigma}}_{\ell'})\right)\\ &= \log L(\widehat{\boldsymbol{\theta}}). \end{aligned}$$

The maximizer $\widehat{\boldsymbol{\theta}}'$ of the lower bound $b(\boldsymbol{\theta})$ in the M-step should satisfy

$$\begin{cases} \left.\dfrac{\partial}{\partial \gamma_\ell} b(\boldsymbol{\theta})\right|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}'} = 0,\\[2ex] \left.\dfrac{\partial}{\partial \boldsymbol{\mu}_\ell} b(\boldsymbol{\theta})\right|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}'} = \boldsymbol{0}_d,\\[2ex] \left.\dfrac{\partial}{\partial \boldsymbol{\Sigma}_\ell} b(\boldsymbol{\theta})\right|_{\boldsymbol{\theta}=\widehat{\boldsymbol{\theta}}'} = \boldsymbol{O}_{d\times d}, \end{cases}$$

from which the maximizer $\widehat{\boldsymbol{\theta}}'$ can be obtained as

$$\begin{cases} \widehat{w}'_\ell = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n}\widehat{\eta}_{i,\ell},\\[2ex] \widehat{\boldsymbol{\mu}}'_\ell = \dfrac{\sum_{i=1}^{n}\widehat{\eta}_{i,\ell}\boldsymbol{x}_i}{\sum_{i'=1}^{n}\widehat{\eta}_{i',\ell}},\\[2ex] \widehat{\boldsymbol{\Sigma}}'_\ell = \dfrac{\sum_{i=1}^{n}\widehat{\eta}_{i,\ell}(\boldsymbol{x}_i-\widehat{\boldsymbol{\mu}}_\ell)(\boldsymbol{x}_i-\widehat{\boldsymbol{\mu}}_\ell)^\top}{\sum_{i'=1}^{n}\widehat{\eta}_{i',\ell}}. \end{cases}$$

The above explanation showed that the log-likelihood is monotone nondecreasing by iterating the E-step and the M-step. Furthermore, the EM algorithm was proved to produce a local optimal solution [120].

A MATLAB code for the EM algorithm is given in Fig. 15.9, and its behavior is illustrated in Fig. 15.10. Here, the mixture model of five Gaussian components is fitted to the mixture of two Gaussian distributions. As shown in Fig. 15.10, two of the five Gaussian components fit the true two Gaussian distributions well, and the remaining three Gaussian components are almost eliminated. Indeed, the learned

```
x=[2*randn(1,100)-5 randn(1,50); randn(1,100) randn(1,50)+3];
[d,n]=size(x);
m=5;
e=rand(n,m);
S=zeros(d,d,m);
for o=1:10000
  e=e./repmat(sum(e,2),[1 m]);
  g=sum(e);
  w=g/n;
  mu=(x*e)./repmat(g,[d 1]);
  for k=1:m
    t=x-repmat(mu(:,k),[1 n]);
    S(:,:,k)=(t.*repmat(e(:,k)’,[d 1]))*t’/g(k);
    e(:,k)=w(k)*det(S(:,:,k))^(-1/2) ...
           *exp(-sum(t.*(S(:,:,k)\t))/2);
  end
  if o>1 && norm(w-w0)+norm(mu-mu0)+norm(S(:)-S0(:))<0.001
    break
  end
  w0=w;
  mu0=mu;
  S0=S;
end

figure(1); clf; hold on
plot(x(1,:),x(2,:),’ro’);
v=linspace(0,2*pi,100);
for k=1:m
  [V,D]=eig(S(:,:,k));
  X=3*w(k)*V’*[cos(v)*D(1,1); sin(v)*D(2,2)];
  plot(mu(1,k)+X(1,:),mu(2,k)+X(2,:),’b-’)
end
```

FIGURE 15.9

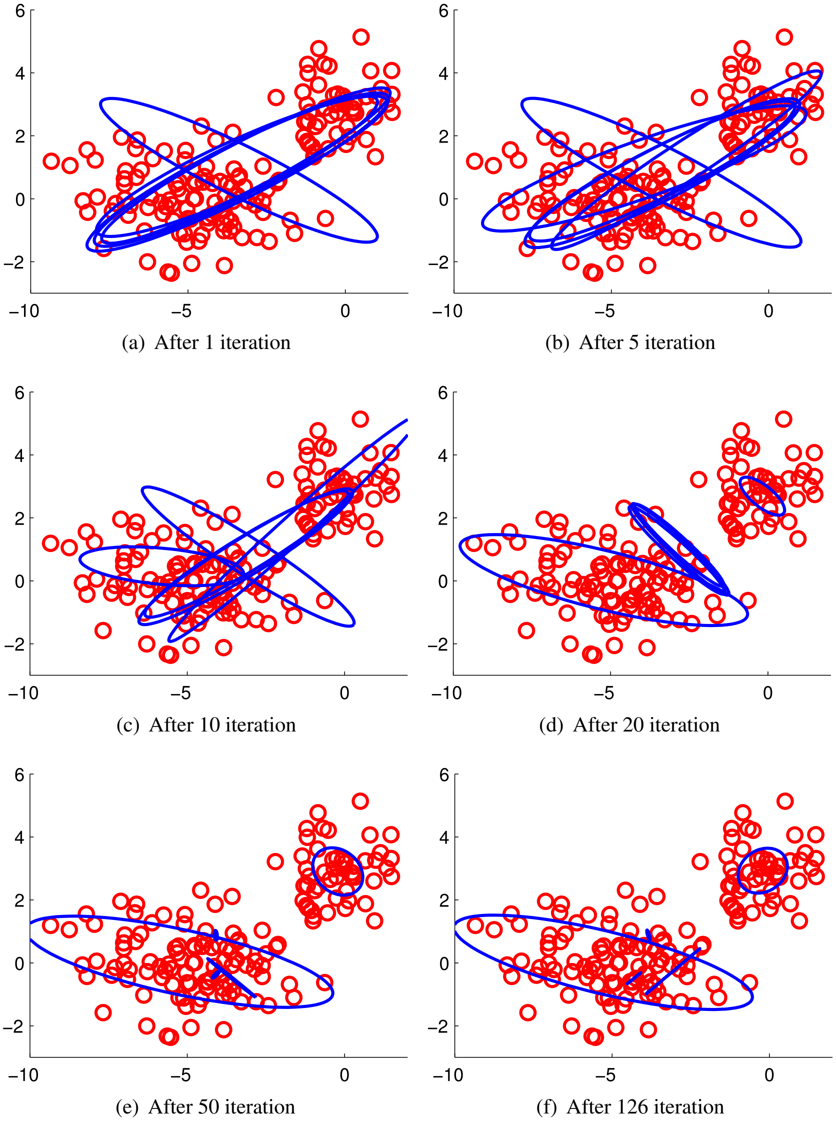
MATLAB code of EM algorithm for Gaussian mixture model.

mixing coefficients are given as

$$(\widehat{w}_1, \widehat{w}_2, \widehat{w}_3, \widehat{w}_4, \widehat{w}_5) = (0.09, 0.32, 0.05, 0.06, 0.49).$$

If the most responsible mixing component,

$$\widehat{y}_i = \operatorname*{argmax}_{\ell} \widehat{\eta}_{i,\ell},$$



(a) After 1 iteration

(b) After 5 iteration

(c) After 10 iteration

(d) After 20 iteration

(e) After 50 iteration

(f) After 126 iteration

FIGURE 15.10

Example of EM algorithm for Gaussian mixture model. The size of ellipses is proportional to the mixing weights $\{w_\ell\}_{\ell=1}^m$.

is selected for each sample $\boldsymbol{x}_i$, density estimation with a mixture model can be regarded as *clustering*. Indeed, the EM algorithm for a Gaussian mixture model is reduced to the *k-means* clustering algorithm for $\boldsymbol{\Sigma}_\ell = \sigma_\ell^2 \boldsymbol{I}$. See Chapter 37 for details.