Chapter 17
Model Based Cluster Analysis of Microarray Data

19

Model-Based Clustering of Microarray Data

19.1 Cluster analysis with the K-means method

The K-means method (MacQueen, 1967; Hartigan, 1975; Hartigan and Wong, 1979) is not a model-based clustering method. It is a data partitioning method that divides the entire data into K disjoint groups. The idea is similar to the multivariate Gaussian mixture model analysis (a model-based method). Both deal with the problem of data partitioning and thus are in clear contrast to the hierarchical clustering methods described in Chapter 18. The K-means method is simple but very useful in microarray data analysis. As a result, it earns a section here in this chapter, even though it is not a model-based clustering method.

In the contest of microarray data analysis, the aim of the K-mean algorithm is to divide all N genes in M dimensions into K clusters so that the within-cluster sum of squares is minimum. It is not practical to require that the solution has minimal within-cluster sum of squares against all partitions, except when N and M are small and $K = 2$. We seek, instead, a "local" optimum, a solution such that no movement of a gene from one cluster to another will reduce the within-cluster sum of squares.

The algorithm requires a data matrix y with N genes in M dimensions (i.e., y is an $M \times N$ matrix) and a matrix of K initial cluster centers (center is also called centroid) denoted by an $M \times K$ matrix μ. The number of genes in cluster k is denoted by N_k for $k = 1, \ldots, K$ and $\sum_{k=1}^{K} N_k = N$. Let $d(j,k)$ be the Euclidean distance between gene j and the centroid of cluster k. The general procedure is to search for a K-partitioning of local minimum within-cluster sum of squares by moving genes from one cluster to another. The within-cluster sum of squares is defined as

$$Q = \sum_{k=1}^{K} \sum_{j=1}^{N_k} d^2(j,k) \tag{19.1}$$

where the Euclidean distance $d(j,k)$ between gene j and the centroid of cluster k is defined as

$$d(j,k) = \|y_j - \mu_k\| = \sqrt{(y_j - \mu_k)^T(y_j - \mu_k)} \tag{19.2}$$

Note that y_j is an $M \times 1$ vector of the expression levels for gene j and μ_k is the centroid (an $M \times 1$ vector) of cluster k.

The K-means algorithm can be summarized as

1. Choose an initial centroid matrix $\mu = \{\mu_1^{(0)}, \ldots, \mu_K^{(0)}\}$
2. Calculate $d(j,k)$ for all $j = 1, \ldots, N$ and $k = 1, \ldots, K$
3. Assign gene j into cluster k if $d(j,k) = \min\{d(j,1), \ldots, d(j,K)\}$
4. When all genes have been assigned, re-calculate $\mu = \{\mu_1, \ldots, \mu_K\}$, where $\mu_k = \frac{1}{N_k}\sum_{j=1}^{N_k} y_j$ is the mean of all y_j that belong to cluster k.
5. Repeat Step 2 to Step 4 until the centroid matrix μ no longer changes. This produces a separation of the genes into K distinct clusters.

The remaining question in the K-means method is how to determine K and the initial centroid matrix μ. The number of clusters is set by the investigator *a priori*. Since the K-means algorithm is computationally fast, one can perform the method using different K values and select the one that produces the most "meaningful" result. Alternatively, the investigator may already have some idea about how many clusters the data should fall into, say K_0. Only a few different K values around K_0 may be evaluated. The initial value of the centroid matrix can be chosen arbitrarily. Since the method only finds a local optimum, different initial values of μ may generate different results. Therefore, multiple initial values of μ should be tried to make sure that the optimum

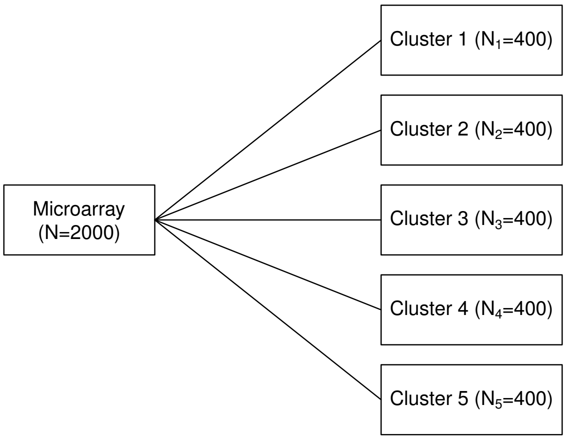

Fig. 19.1. K-means clustering analysis for $N = 2000$ genes with $K = 5$ clusters each having $N_k = 400$ genes.

obtained is close to the global optimum. One approach to choosing the initial value of μ is to randomly partition the data into K clusters and use the mean value of each cluster to form the centroid matrix μ. Another approach is to hand-pick K genes which appear to be very "different" and use the expression levels of these genes as the centroids. The K-means method that updates the centroids constantly based on the expression levels of the genes included in each cluster is called unsupervised K-means. The final centroid matrix μ is entirely determined by the data.

In some applications of the K-means method, the entire set of genes are divided into two categories. In one category, the genes are already assigned *a priori* into K clusters based on prior knowledge of the investigators. The other category contains genes whose cluster identities are not known. The purpose of the K-mean analysis is to assign each of the genes in category two to one of the K clusters defined by genes in category one. This type of analysis is called supervised cluster analysis. Genes in category one are called the training sample while genes in category two are called the testing sample. The training sample is only used to provide the initial centroid matrix $\mu^{(0)}$. Let S_k be the number of genes in cluster k of the training sample for $k = 1, \ldots, K$ and $\sum_{k=1}^{K} S_k = S$, where S is the total number of genes in the training sample. In the supervised K-means analysis, the updated centroid matrix is calculated using

$$\mu_k = \frac{1}{N_k + S_k} \left(S_k \mu^{(0)} + \sum\nolimits_{j=1}^{N_k} y_j \right) \tag{19.3}$$

where j now indexes all genes in cluster k of the testing sample. The new μ_k will never replace $\mu^{(0)}$ but it is updated constantly using the above equation. The iteration process stops when further moving genes from one cluster to another does not change the centroid. Figure 19.1 illustrates schematically the result of a K-means clustering analysis for $N = 2000$ genes with $K = 5$ clusters each having $N_k = 400$ genes.

In contrast to the hierarchical clustering analysis, which can only handle a few hundred genes at a time, the K-means method, along with the model-based methods to be introduced later, can handle almost unlimited number of genes.

19.2 Cluster analysis under Gaussian mixture

In contrast to the hierarchical clustering methods, the model-based clustering analysis classifies genes into distinct clusters. Among clusters, genes show different patterns of expression but within clusters genes share the same patterns of expression. The method requires specific statistical models to fit the data and the errors of fitness are supposed to follow some specific distributions. Theory and method of Gaussian mixture can be found in McLachlan and Peel (2000) and Fraley and Raftery (2002). The Gaussian mixture models have been widely applied to microarray data analysis (Ghosh and Chinnaiyan, 2002; McLachlan et al., 2002; Pan et al., 2002; Ouyang et al., 2004; Qu and Xu, 2004, 2006; McNicholas and Murphy, 2010).

19.2.1 Multivariate Gaussian distribution

Before we introduce the mixture model of microarray data analysis, we will review the the basic definition of multivariate normal distribution, which is also called multivariate Gaussian distribution. Let us denote the expression of the jth gene across M samples by a column vector y_j. Assume that y_j follows a multivariate Gaussian distribution denoted by

$$y_j \sim N(\mu, \Sigma) \tag{19.4}$$

where μ is an $M \times 1$ vector of means (not the centroid matrix appearing in the K-means method) and Σ is an $M \times M$ symmetrical and positive definite matrix expressed as

$$\Sigma = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \dots & \sigma_{1M} \\ \sigma_{12} & \sigma_{22} & \dots & \sigma_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{1M} & \sigma_{2M} & \dots & \sigma_{MM} \end{bmatrix} \tag{19.5}$$

The multivariate normal density is

$$\phi(y_j|\mu, \Sigma) = \frac{1}{(2\pi)^{M/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(y_j - \mu)^T \Sigma^{-1}(y_j - \mu)\right] \tag{19.6}$$

To estimate the parameters, $\theta = \{\mu, \Sigma\}$, we construct the following log likelihood function,

$$L(\theta) = -\frac{N}{2}\log|\Sigma| - \frac{1}{2}\sum_{j=1}^{N}(y_j - \mu)^T \Sigma^{-1}(y_j - \mu) \tag{19.7}$$

The parameters can be estimated using the maximum likelihood method by setting

$$\frac{\partial}{\partial\mu}L(\theta) = \frac{\partial}{\partial\Sigma}L(\theta) = 0 \tag{19.8}$$

and solving for θ. This requires rules for matrix derivatives (beyond the scope of this class), but the final result for μ is

$$\begin{aligned} \frac{\partial}{\partial\mu}L(\theta) &= -\frac{1}{2}\frac{\partial}{\partial\mu}\sum_{j=1}^{N}(y_j - \mu)^T \Sigma^{-1}(y_j - \mu) \\ &= \Sigma^{-1}\sum_{j=1}^{N}(y_j - \mu) \\ &= \Sigma^{-1}\sum_{j=1}^{N} y_j - N\Sigma^{-1}\mu \end{aligned} \tag{19.9}$$

Setting $\frac{\partial}{\partial \mu} L(\theta) = 0$ and solving for μ leads to

$$\hat{\mu} = \frac{1}{N} \sum_{j=1}^{N} y_j \tag{19.10}$$

The partial derivative of the log likelihood function with respect to Σ is

$$\begin{aligned} \frac{\partial}{\partial \Sigma} L(\theta) &= -\frac{N}{2} \frac{\partial}{\partial \Sigma} \log |\Sigma| - \frac{1}{2} \frac{\partial}{\partial \Sigma} \sum_{j=1}^{N} (y_j - \mu)^T \Sigma^{-1} (y_j - \mu) \\ &= -\frac{N}{2} \frac{1}{|\Sigma|} \frac{\partial}{\partial \Sigma} |\Sigma| - \frac{1}{2} \frac{\partial}{\partial \Sigma} \mathrm{tr} \left[\Sigma^{-1} \sum_{j=1}^{N} (y_j - \mu)(y_j - \mu)^T \right] \\ &= -\frac{N}{2} \Sigma^{-1} + \frac{1}{2} \Sigma^{-1} \sum_{j=1}^{N} (y_j - \mu)(y_i - \mu)^T \Sigma^{-1} \end{aligned} \tag{19.11}$$

Setting $\frac{\partial}{\partial \Sigma} L(\theta) = 0$ and solving for Σ yields

$$\hat{\Sigma} = \frac{1}{N} \sum_{j=1}^{N} (y_j - \hat{\mu})(y_j - \hat{\mu})^T \tag{19.12}$$

The MLE of the parameters can be solved explicitly without resorting to any iterative algorithm.

19.2.2 Mixture distribution

Assume that these genes are sampled from K multivariate normal distributions (clusters), but we do not know which gene comes from which cluster. The problem becomes a mixture model problem. Let $G_j = k$ be the cluster index for the jth gene, i.e., if the jth gene is from the kth cluster, then $G_j = k, \forall k = 1, \ldots, K$. Let us redefine $\mu = \{\mu_1, \ldots, \mu_K\}$ as an $M \times K$ matrix (similar to the centroid in the K-means analysis), where μ_k is an $M \times 1$ vector for the mean of cluster k. Let us define Σ_k as the covariance matrix of cluster k. Given that the jth gene is from the kth cluster, the density of y_j is multivariate normal with mean μ_k and variance matrix Σ_k, i.e., $y_j | G_j = k \sim N(\mu_k, \Sigma_k), \forall k = 1, \ldots, K$. The conditional density of y_j given $G_j = k$ is

$$\phi_k(y_j | \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{M/2} |\Sigma_k|^{1/2}} \exp \left[-\frac{1}{2} (y_j - \mu_k)^T \Sigma_k^{-1} (y_j - \mu_k) \right] \tag{19.13}$$

Let $\pi_k > 0$ be the proportion of genes belonging to the kth cluster, where $\sum_{k=1}^{K} \pi_k = 1$. These proportions are also called the mixing proportions. They are treated as the prior probability that a randomly sampled gene is from the kth cluster. The density of the mixture distribution is

$$f(y_j|\theta) = \sum_{k=1}^{K} \pi_k \phi_k(y_j|\mu_k, \Sigma_k) \tag{19.14}$$

where

$$\theta = \{\pi_1, ..., \pi_K, \mu_1, ..., \mu_K, \Sigma_1, ..., \Sigma_K\} \tag{19.15}$$

is the vector of parameters. The overall log likelihood function is

$$L(\theta) = \sum_{j=1}^{N} \ln f(y_j|\theta) \tag{19.16}$$

The likelihood function for the mixture model is messy and thus no explicit solution for θ is available. Therefore, we resort to a numerical algorithm for estimating the MLE of θ.

19.2.3 The EM algorithm

The EM algorithm for parameter estimation largely remains the same as that in the segregation analysis except that the dimension of parameters becomes multivariate. Instead of re-deriving the EM algorithm, we simply provide detailed steps of the EM algorithm. With the EM algorithm, we assign each gene into one of the K clusters with a certain probability denoted by

$$\pi_{jk} = \Pr(G_j = k|y_j, \theta) \tag{19.17}$$

This probability is called the posterior probability of $G_j = k$ given the data and the parameter values. The EM algorithm starts with some initial values of all unknown parameters and iteratively update each parameter conditional on the parameter values in the previous round of iteration. The iteration process is summarized as follows.

1. Set $t = 0$ and initialize all parameters $\theta^{(t)}$, including

$$\pi_k^{(t)} = 1/K, \forall k = 1, \ldots, K \tag{19.18}$$

2. Update the posterior probabilities of cluster assignments,

$$\pi_{jk}^{(t)} = \frac{\pi_k^{(t)} \phi_k(y_j|\mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{k'=1}^{K} \pi_{k'}^{(t)} \phi_{k'}(y_j|\mu_{k'}^{(t)}, \Sigma_{k'}^{(t)})} \tag{19.19}$$

for all $j = 1, \ldots, N$ & $k = 1, \ldots, K$

3. Update the mean vectors by

$$\mu_k^{(t+1)} = \left(N\pi_k^{(t)}\right)^{-1} \sum_{j=1}^{N} \pi_{jk}^{(t)} y_j, \forall k = 1, \dots, K \tag{19.20}$$

4. Update the covariance matrices by

$$\Sigma_k^{(t+1)} = \left(N\pi_k^{(t)}\right)^{-1} \sum_{j=1}^{N} \pi_{jk}^{(t)} \left(y_j - \mu_k^{(t+1)}\right) \left(y_j - \mu_k^{(t+1)}\right)^T \tag{19.21}$$

   for all $k = 1, \dots, K$
5. Update the mixing proportions,

$$\pi_k^{(t+1)} = \frac{1}{N} \sum_{j=1}^{N} \pi_{jk}^{(t)}, \forall k = 1, \dots, K \tag{19.22}$$

6. Increment t by $t + 1$ and repeat Steps 2 to Step 5 until a certain criterion of convergence is reached.

Once the EM iteration converges, the parameter values are the maximum likelihood estimates, denoted by $\hat{\theta}$. The conditional posterior probability of gene j from cluster k is denoted by $\hat{\pi}_{jk}$. Gene j will be assigned to the kth cluster if

$$\hat{\pi}_{jk} = \max\{\hat{\pi}_{j1}, ..., \hat{\pi}_{jK}\} \tag{19.23}$$

One can also set a cutoff point for cluster assignment. For example, assign gene j into cluster k if $\hat{\pi}_{jk} = \max\{\hat{\pi}_{j1}, ..., \hat{\pi}_{jK}\} > \alpha$, where $0 < \alpha < 1$ is the cutoff point. Those genes whose maximum posterior probability is less than α are claimed to be unassigned. This flexibility of the model-based method is an advantage over the K-means method.

19.2.4 Supervised cluster analysis

In the supervised cluster analysis, we know the functions of genes in the training sample, and thus know which gene belongs to which cluster in the training sample. The purpose of the supervised cluster analysis is to assign genes in the testing sample (contains genes with unknown functions) into one of the clusters defined in the training sample. The method was developed by Qu and Xu (2004). Let S be the training sample size, which is partitioned into K clusters with the sample size for the kth cluster being S_k, for $\sum_{k=1}^{K} S_k = S$. From the training sample, we estimate $\tilde{\mu} = \{\tilde{\mu}_1, \dots, \tilde{\mu}_K\}$ and $\widetilde{\Sigma} = \{\widetilde{\Sigma}_1, \dots, \widetilde{\Sigma}_K\}$, where $\tilde{\mu}_k$ is an $M \times 1$ vector for the mean of cluster k and $\widetilde{\Sigma}_k$ is an $M \times M$ variance-covariance matrix for cluster k. The estimated $\tilde{\mu}$ and $\widetilde{\Sigma}$ obtained from the training sample are used to guide the cluster analysis for the testing sample. Equations (19.10) and (19.12) introduced in a previous section are used to estimate $\tilde{\mu}_k$ and $\widetilde{\Sigma}_k$, respectively, except that the sample size N in those equations is now replaced by S_k because the sample size for cluster k in the training sample is S_k. This concludes the first step of the supervised cluster analysis.

The second step of the supervised cluster analysis is to update the estimated parameters from genes in the testing sample and assign each of the N genes in the testing sample to one of the K clusters. We will again use the EM algorithm to update the parameters, which is described as follows,

1. Set $t = 0$ and initialize all parameters, including
$$\mu_k^{(t)} = \tilde{\mu}_k,\ \Sigma_k^{(t)} = \widetilde{\Sigma}_k \text{ and } \pi_k^{(t)} = \frac{S_k + N/K}{S+N}, \forall k = 1, \ldots, K$$
2. Update the posterior probabilities of cluster assignments,
$$\pi_{jk}^{(t)} = \frac{\pi_k^{(t)} \phi_k(y_j | \mu_k^{(t)}, \Sigma_k^{(t)})}{\sum_{k'=1}^K \pi_{k'}^{(t)} \phi_{k'}(y_j | \mu_{k'}^{(t)}, \Sigma_{k'}^{(t)})} \tag{19.24}$$
for all $\forall j = 1, \ldots, N$ & $k = 1, \ldots, K$
3. Update the mean vectors by
$$\mu_k^{(t+1)} = \left[S_k + \sum_{j=1}^N \pi_{jk}^{(t)} \right]^{-1} \left[S_k \tilde{\mu}_k + \sum_{j=1}^N \pi_{jk}^{(t)} y_j \right] \tag{19.25}$$
for all $k = 1, \ldots, K$
4. Update the variance-covariance matrices by
$$\begin{aligned} \Sigma_k^{(t+1)} = & \left[S_k + \sum_{j=1}^N \pi_{jk}^{(t)} \right]^{-1} \\ & \times \left[S_k \widetilde{\Sigma}_k + \sum_{j=1}^N \pi_{jk}^{(t)} (y_j - \mu_k^{(t+1)})(y_j - \mu_k^{(t+1)})^T \right] \end{aligned} \tag{19.26}$$
for all $k = 1, \ldots, K$
5. Update the mixing proportions,
$$\pi_k^{(t+1)} = \frac{1}{S+N} \left[S_k + \sum_{j=1}^N \pi_{jk}^{(t)} \right], \forall k = 1, \ldots, K \tag{19.27}$$
6. Increment t by $t + 1$ and repeat Steps 2 to 5 until a certain criterion of convergence is reached.

19.2.5 Semi-supervised cluster analysis

It is possible that genes in the training sample may not cover all possible clusters in the testing sample. In other words, some genes in the testing sample may not belong to any of the clusters in the training sample. In this case, the supervised cluster analysis may be combined with the unsupervised cluster analysis, which is called the semi-supervised cluster analysis. Assume that we want to classify all N genes into K clusters. Among the K clusters, $K' < K$ of them are defined in the training sample where $K - K'$ clusters occur only in the testing sample. The algorithm is identical to that of the supervised algorithm except that $S_k = 0$ for $k = K' + 1, \ldots, K$. In other words, we set each of the the additional clusters that are not included in the training sample empty. No further modification is required. The values of $\tilde{\mu}_k$ and $\widetilde{\Sigma}_k$ for $k = K' + 1, \ldots, K$ are arbitrary, e.g., $\tilde{\mu}_k = 0$ and $\widetilde{\Sigma}_k = 0$, because they do not affect the EM estimates of the parameters.

19.3 Inferring the number of clusters

The number of clusters K is usually unknown. The EM algorithm described above is based on a fixed value of K. In the unsupervised and semi-supervised methods, the K value needs to be estimated from the data. Note that in the supervised cluster analysis, K is determined exclusively by the training sample and thus it is given. In this section, we will introduce a special method to infer K, called the Bayesian information criterion (BIC), which was proposed by Schwarz (1978), although there are many other methods available in the literature. For the unsupervised cluster analysis, the BIC for K clusters is given by

$$\Lambda(K) = -2L(\theta|K) + \dim(\theta|K)\ln(N) \tag{19.28}$$

where

$$L(\theta|K) = \sum_{j=1}^{N} \ln \sum_{k=1}^{K} \pi_k \phi_k(y_j|\mu_k, \Sigma_k) \tag{19.29}$$

is the log likelihood function evaluated at $\theta = \hat{\theta}$ and

$$\dim(\theta|K) = MK + \frac{1}{2}M(M+1)K + K - 1 \tag{19.30}$$

is the dimension of parameter vector θ. Recall that

$$\theta = \{\mu_1, \ldots, \mu_K, \Sigma_1, \ldots, \Sigma_k, \pi_1 \ldots, \pi_K\}$$

where the dimension for the μ's is MK, the dimension for the Σ's is $\frac{1}{2}M(M+1)K$ and the dimension for the π's is $K-1$. The $K-1$ comes from the fact that only $K-1$ of the π's are independent parameters because $\pi_K = 1 - \sum_{k=1}^{K-1} \pi_k$.

Using the BIC as the criterion of optimality, the K value that minimizes $\Lambda(K)$ is the optimal number of clusters.

The BIC for the semi-supervised clustering for K clusters is also given by equation (19.28). The dimension of the parameters remains the same as that given in equation (19.30). However, the log likelihood function is different from equation (19.29). The correct log likelihood function for the semi-supervised clustering is

$$\begin{aligned} L(\theta|K) = & \sum_{j=1}^{N} \ln \sum_{k=1}^{K} \pi_k \phi_k(y_j|\mu_k, \Sigma_k) \\ & + \sum_{k=1}^{K} \sum_{j=1}^{S_k} \ln \phi_k(y_j^*|\mu_k, \Sigma_k) \end{aligned} \tag{19.31}$$

where y_j^* indicates data from the training sample. The additional information gained from the training sample appears in the second term of the right hand side of Equation (19.31).

19.4 Microarray experiments with replications

Recall that in the microarray experiment without replication,

$$y_j = \{y_{1j}, \dots, y_{Mj}\}$$

is an $M \times 1$ vector for the expression level of gene j in all M conditions. Let us assume that each condition represents a subject (e.g., a human, an animal or a plant). Assume that the ith subject of the experiment is replicated r_i times and the mean value of the r_i replications is entered into the data set. In this case, the ith row and the jth column of matrix y is actually $\bar{y}_{ij}$, a mean of of r_i replications. We now use a linear model to describe a single measurement y_{ij} by

$$y_{ij} = \gamma_{ij} + \epsilon_{ij} \tag{19.32}$$

where γ_{ij} is the true expression of subject i for gene j and ϵ_{ij} is a measurement error with an assumed $N(0, \sigma^2)$ distribution. If $\bar{y}_{ij}$ is the mean of r_i replications, the model becomes

$$\bar{y}_{ij} = \gamma_{ij} + \bar{\epsilon}_{ij} \tag{19.33}$$

where $\bar{\epsilon}_{ij}$ is the mean error with an $N(0, \sigma^2/r_i)$ distribution. We now assume that vector $\gamma_j = \{\gamma_{1j}, \dots, \gamma_{Mj}\}$ has a mixture of K multivariate Gaussian distributions with the kth component (cluster) being $\gamma_j \sim N(\mu_k, \Sigma_k)$. Now let us define $\bar{\epsilon}_j = \{\bar{\epsilon}_{1j}, \dots, \bar{\epsilon}_{Mj}\}$ as a vector of the mean errors and $\bar{y}_j = \{\bar{y}_{1j}, \dots, \bar{y}_{Mj}\}$ as a vector for the mean expression of gene j. We can write the following model to describe

$$\bar{y}_j = \gamma_j + \bar{\epsilon}_j \tag{19.34}$$

where $\bar{\epsilon}_j \sim N(0, D\sigma^2)$ and D is an $M \times M$ diagonal matrix,

$$D = \begin{bmatrix} \frac{1}{r_1} & 0 & \dots & 0 \\ 0 & \frac{1}{r_2} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{r_M} \end{bmatrix} \tag{19.35}$$

This leads to a multivariate Gaussian mixture distribution for $\bar{y}_j$,

$$\bar{y}_j \sim \sum_{k=1}^{K} \pi_k N(\mu_k, \Sigma_k + D\sigma^2) \tag{19.36}$$

Comparing this distribution to the multivariate Gaussian distribution in the unreplicated experiment introduced before, we can see that there is an extra parameter σ^2 involved, which represents the variance of repeated measurement errors. The parameter vector now is defined as

$$\theta = \{\pi_1, \dots, \pi_K, \mu_1, \dots, \mu_K, \Sigma_1, \dots, \Sigma_K, \sigma^2\}$$

The log likelihood function is

$$L(\theta) = \sum_{j=1}^{N} \ln \sum_{k=1}^{K} \pi_k \phi_k(\bar{y}_j | \mu_k, \Sigma_k + D\sigma^2) \tag{19.37}$$

The EM algorithm for the MLE of parameters is summarized below.

1. Set $t = 0$ and initialize all parameters $\theta^{(t)}$, including $\sigma^{2(t)}$ and
$$\pi_k^{(t)} = 1/K, \forall k = 1, \dots, K$$
2. Update the posterior probabilities of cluster assignments,
$$\pi_{jk}^{(t)} = \frac{\pi_k^{(t)} \phi_k(\bar{y}_j | \mu_k^{(t)}, \Sigma_k^{(t)} + D\sigma^{2(t)})}{\sum_{k'=1}^{K} \pi_{k'}^{(t)} \phi_{k'}(\bar{y}_j | \mu_{k'}^{(t)}, \Sigma_{k'}^{(t)} + D\sigma^{2(t)})} \tag{19.38}$$
for all $j = 1, \dots, N$ & $k = 1, \dots, K$
3. Update the mean vectors using
$$\mu_k^{(t+1)} = \left[\sum_{j=1}^{N} \pi_{jk}^{(t)} V_k^{-1} \right]^{-1} \left[\sum_{j=1}^{N} \pi_{jk}^{(t)} V_k^{-1} \bar{y}_j \right] \tag{19.39}$$
where $V_k = \Sigma_k + D\sigma^{2(t)}, \forall k = 1, \dots, K$.

4. Update the covariance matrices by

$$\varSigma_k^{(t+1)} = \left(N\pi_k^{(t)}\right)^{-1} \sum_{j=1}^{N} \pi_{jk}^{(t)} \mathtt{E}\left[(\gamma_j - \mu_k^{(t+1)})(\gamma_j - \mu_k^{(t+1)})^T\right] \tag{19.40}$$

for all $k = 1, \ldots, K$

5. Update σ^2 by

$$\sigma^{2(t+1)} = \frac{1}{MN} \sum_{j=1}^{N} \sum_{k=1}^{K} \pi_{jk}^{(t)} \mathtt{E}\left[(\bar{y}_j - \gamma_j)^T D^{-1} (\bar{y}_j - \gamma_j)\right] \tag{19.41}$$

6. Update the mixing proportions by

$$\pi_k^{(t+1)} = \frac{1}{N} \sum_{j=1}^{N} \pi_{jk}^{(t)}, \forall k = 1, \ldots, K \tag{19.42}$$

7. Increment t by $t+1$ and repeat Steps 2 to 6 until a certain criterion of convergence is reached.

In the above EM iteration process, γ_j appears twice, all in the expected quadratic forms. The first appearance is

$$\mathrm{E}\left[(\gamma_j - \mu_k)(\gamma_j - \mu_k)^T\right] = \mathrm{E}(\gamma_j - \mu_k)\mathrm{E}(\gamma_j - \mu_k)^T + \mathrm{var}(\gamma_j - \mu_k) \tag{19.43}$$

and the second appearance is

$$\mathrm{E}\left[(\gamma_j - \bar{y}_j)^T D^{-1} (\gamma_j - \bar{y}_j)\right] = \\ \mathrm{E}(\gamma_j - \bar{y}_j)^T D^{-1} \mathrm{E}(\gamma_j - \bar{y}_j) + \mathrm{tr}\left[D^{-1}\mathrm{var}(\gamma_j - \bar{y}_j)\right] \tag{19.44}$$

Although the cluster label k does not occur explicitly in Equation (19.44), this equation is cluster specific. For the kth cluster, we have

$$\mathrm{E}(\gamma_j - \bar{y}_j) = \mathrm{E}(\gamma_j - \mu_k) + \mathrm{E}(\mu_k - \bar{y}_j) = \mathrm{E}(\gamma_j - \mu_k) + (\mu_k - \bar{y}_j) \tag{19.45}$$

and

$$\mathrm{var}(\gamma_j - \bar{y}_j) = \mathrm{var}(\gamma_j - \mu_k) + \mathrm{var}(\mu_k - \bar{y}_j) = \mathrm{var}(\gamma_j - \mu_k) \tag{19.46}$$

This result is due to $\mathrm{E}(\mu_k - \bar{y}_j) = (\mu_k - \bar{y}_j)$ and $\mathrm{var}(\mu_k - \bar{y}_j) = 0$ as $\mu_k - \bar{y}_j$ is not a function of variable γ_j. Now there are only two terms that need our attention, which are $\mathrm{E}(\gamma_j - \mu_k)$ and $\mathrm{var}(\gamma_j - \mu_k)$. They are the posterior mean and posterior variance of γ_j for cluster k given below,

$$\mathrm{E}(\gamma_j - \mu_k) = \varSigma_k(\varSigma_k + D\sigma^2)^{-1}(\bar{y}_j - \mu_k) \tag{19.47}$$

and

$$\mathrm{var}(\gamma_j - \mu_k) = \Sigma_k - \Sigma_k(\Sigma_k + D\sigma^2)^{-1}\Sigma_k \tag{19.48}$$

The optimal number of clusters can be found using the BIC criterion (see Equation 19.29). The dimension of the parameters, however, is

$$\dim(\theta|K) = MK + \frac{1}{2}M(M+1)K + K \tag{19.49}$$

which is one shy that of the experiment without replication.

One caveat of the model based cluster analysis is the "identifiability" problem, which occurs as two or more clusters having identical distributions (i.e., same cluster mean and the same cluster variance matrix). Several approaches can be used to solve this problem. One *ad hoc* approach (adopted by Qu and Xu (2006)) is to introduce a small noise vector to each μ_k at each iteration. This small perturbation will eventually separate all μ_k from each other. Another approach is to revise the model so that all clusters share the same variance matrix, i.e., $\Sigma_k = \Sigma, \forall k = 1, \ldots, K$. The most effective approach is the stochastic EM algorithm (SEM), in which the cluster label for each gene is randomly sampled from the posterior distribution rather than taking the posterior mean (Zhan et al., 2011). The SEM algorithm will be described in the next chapter (Chapter 20). Conceptually, this approach is the same as the *ad hoc* method, but statistically it is more rigorous and should generate the best result of clustering.