Model-based Clustering

by

HaiJiang Steven Shi

A research paper

presented to the University of Waterloo

in fulfilment of the

research paper requirement for the degree of

Master of Mathematics

in

Statistical Computing

Waterloo, Ontario, Canada, 2005

AUTHOR'S DECLARATION FOR ELECTRONIC SUBMISSION OF A Research Paper

I hereby declare that I am the sole author of this research paper. This is a true copy of the research, including any required final revisions, as accepted by my examiners.

I understand that my research paper may be made electronically available to the public.

Abstract

Sokal et al. proposed cluster analysis in the late 1950s. It is a method to find the cohesive groups based on measured characteristics using numerical measurement. Typical clustering methods are: partitioning methods, hierarchical methods, density-based methods, grid-based methods and model-based methods. My research is focus on model-based clustering. Here, only Multivariate Normal Distribution is considered.

Parameters in Multivariate Normal Distribution are considered by geometrically decomposition of the covariance matrix. Two different approaches for the parameters estimate are EM (Expectation-Maximization) and Gibbs Sampler. Both are based on Bayesian Theorem by introducing class label vector as latent variable.

Two model-selection approaches to solve the posterior integrated likelihood problem for Bayesian factor, which determining the best model from a list of candidate models, are BIC and Laplace approximation.

List of Tables

Table of Contents

1. Introduction

Data mining has become very popular for many years. Its goal is to extract information from any data source. If the data source is database, then it is traditional database mining, which is mostly investigated based on the full and clear data structure. If the data source is text file, it is called text mining, which is also studied for years. As an unstructured database, both variable and noise need to be furthering handled before any statistical algorithm is carried. Lots of methods have been involved in this area.

Before any data mining methods are applied, we need to understand where the data comes from. Some are from experiments, which mean we can control the outcome variable by changing different input variable group. That is not the area of data mining because we already know the data enough. Others are from observation, which means the data was collected as it is, and there is not way to make the data. Here, only this kind of out of control dataset was considered.

1.1 Supervised vs. Unsupervised learning

From the problem itself, data mining can be classified into two areas: supervised learning and unsupervised learning. Supervised learning, which is also called classification problem, uses information from training dataset including class labels to find the classification rule to classify test data set. In order to avoid the problem of over fitting, a small part of data (Say, 10 percent), which is separated from the training data set, is used to validate the performance of classification rule. Unsupervised learning, which is also called clustering method, uses the whole data set without class labels. Its goal is to find the criterion to divide the data into several groups, with the observation within each group share some common attributes. Usually, it is hard to evaluate the performance of the clustering result.

Data mining is closely related to two disciplines: exploratory data analysis in statistics and knowledge discovery & machine learning in computer science. It is not a simple way to apply a statistical model, since the statistical models are usually based on many assumptions.

The most popular classification methods are CART (classification and regression tree), Neural Network, kNN (k nearest neighborhood), Logistic regression, Naive Bayes, and etc. The classification performance can be evaluated using industrial standard, such as precision, Recall, and F-measure.

1.2 Clustering analysis

Sokal, Sneath, and others proposed cluster analysis in the late 1950s. It is a method to find the cohesive groups based on measured characteristics using numerical measurement.

Researches are mainly based on a set of heuristic methods, such as partitioning method and hierarchical clustering. Typical partitioning methods are K-means and etc. Partitioning method is usually based on dissimilarity measurement between observations, which is often used together with hierarchical method. The distance criterion, which is used to measure the dissimilarity, includes Manhattan distance and Euclidean distance.

As a very straightforward approach, it is hard to know how many clusters we need, how to compare the performance between methods, and there is no way to deal with outliers in heuristic methods.

In order to better understand clustering performance, we need to use probability model to access all these uncertainty. Whether this method is likely to be better than others or can suggest better methods.

Model-based clustering method is based on probability model from the data. We assume data are come from some distribution functions. So the reason to divide the data into the two groups is that the data come from a mixture of two different probability models.

1.3 Model-based Clustering

MLE (maximum likelihood estimation) is used in model-based clustering method to find the parameter inside the probability model. Since the probability function is a mixture summation of a couple of probability function, it makes the traditional method infeasible to find the maximum value. Latent variable technique is used here, relocation algorithm such as EM and Gibbs sampling are among the most popular.

The criterion to split one data set into several data sets is to make the variance between the clusters maximum and inside the clusters minimum.

The following sections are organized as follows: Section 2 gives a brief introduction of the mathematical model, which brought us the corresponding question. Section 3 gave a general solution of EM with parameter estimate. Section 4 gave a general solution of Gibbs Sampler regarding to parameter estimate. Section 5 gave the detail procedure for Model-based clustering involving two approaches for both model selection and clustering procedure. And some useful packages were given as model-based clustering tools.

2. Mathematical Model

2.1 Finite mixture model

Assume input data $\underline{x}_1, \underline{x}_2, ..., \underline{x}_n$ is a random sample from an unknown distribution function with n observations in p dimensional space. The entire sample X is defined as $\left(\underline{x}_1^{\,T}, \underline{x}_2^{\,T}, ..., \underline{x}_n^{\,T}\right)^T$, where the superscript T denotes vector transpose.

In order to model the data, a mixture model is very useful in practice. The observations are often heterogeneous, rather than one single homogeneous group, and can often by modeled by a mixture distribution. Assume that there is only finite number of clusters in the model, that the number of clusters is fixed, and that data are from different clusters. In each cluster, data is assumed coming from some probability distributions.

A finite mixture distribution is a weighted linear combination of a finite number of simple component distributions:

$$\sim \sum_{k=1}^{g} \left\{\pi_k \bullet f_k\left(\underline{x}_i; \underline{\theta}_k,\right)\right\}$$

where the vector Θ containing all unknown parameters in the mixture model, and can be written as $\left\{\pi_1, ..., \pi_{g-1}, \theta_1, ..., \theta_g\right\}$; g is the number of components. The component probabilityπ_k represents the probability that an observation will come from the kth component, and so lies in between 0 and 1, and sums to 1, $f_k\left(\underline{x}_i; \theta_k,\right)$ is the kth component distribution function and θ_k is the kth component parameter.

The component distributions may or may not have the same form. Theoretically, it can be any form of distribution including Bernoulli, Poisson, Normal, t and etc. The multivariate normal distribution is often used as the common mixture component.

If the component distribution is normal, the finite mixture model becomes the finite multivariate normal mixture model. The main difficulty of these models is to determine in advance which distributional forms should be used for the particular problem. Probabilistic model-based clustering methods assume a probability model for each cluster, and so are best way when we have reason to believe that the component distribution forms are appropriate.

2.2 Finite multivariate normal mixture model

Suppose the component distribution is multivariate normal in p dimensional space with mean vector $\underline{u_k}$ and covariance matrix $\sum_k$; that is

$$f_{Mixture}\left(\underline{x_i} \mid \Theta\right) = \sum_{k=1}^{g} \pi_k \phi\left(\underline{x_i} \mid \underline{u_k}, \Sigma_k\right)$$

where component parameters $\underline{\theta_k}$ have become $\left(\underline{u_k}; \Sigma_k\right)$ and parameter Θ is

$$\left(\pi_1, \ldots, \pi_{g-1}, \underline{u_1}, \ldots, \underline{u_g}, \Sigma_1, \ldots, \Sigma_g\right)$$

In the case of the multivariate normal, the relationship between observations can be measured by covariance matrix. The covariance matrix in the kth group can be modeled parsimoniously in a geometrically interpretable way using a variant of the standard spectral decomposition of a covariance matrix, namely:

$$\sum_k = \lambda_k O_k D_k O_k^T \quad \text{(Covariance Matrix)}$$

where λ_k is a scalar constant, and represents the volume of the kth covariate matrix; O_k is an orthogonal matrix representing the orientation of the kth covariate matrix; D_k

is a diagonal matrix, represents the shape of the kth covariate matrix, with the form $Diag\{\alpha_{1k},\alpha_{2k},..,\alpha_{pk}\}$, where α_{1k} >= α_{2k} >= α_{3k} >= .. >= α_{pk} >= 0 . This is the covariance matrix representation given in Banfield and Raftery (1993).

This decomposition makes the covariance matrix easy to understand. Banfield and Raftery (1993) consider eight possibilities as shown in Table 1. These determine different covariance structure, which are easily interpreted geometrically via this decomposition. They range for from a simple diagonal in structure 1 with spherical shape and the same volume to the absence of any common covariance in structure 8.

Table 1: Covariate matrix decomposition for geometric interpretation

Structure	Σ_k	D (Shape)	O (Orientation)	λ (Volume)
1	λI	Spherical	N/A	Same
2	$\lambda_k I$	Spherical	N/A	Different
3	Σ	Same	Same	Same
4	$\lambda_k \sum_0$	Same	Same	Different
5	$\lambda O_k D O_k^T$	Same	Different	Same
6	$\lambda_k O_k D O_k^T$	Same	Different	Different
7	$\lambda_k O D_k O^T$	Different	Same	Different
8	Σ_k	Different	Different	Different

2.3 Estimation

In order to estimate the parameters of the mixture distribution, many approaches have been developed, such as graphical methods, method of moments, minimum-distance methods, maximum likelihood, and Bayesian approaches. But explicit formulas for the parameter estimate are typically not available (McLachlan & Peel).

Maximum likelihood estimation method has been by far the most commonly used approach to the fitting of mixture distributions with the likelihood function.

$$L(\Theta;\ X)\propto\prod_{i=1}^{n} f_{Mixture}\left(\underline{x_i}\,|\,\underline{\theta_k}\right)=\prod_{i=1}^{n}\sum_{k=1}^{g}\pi_k f_k\left(\underline{x_i}\,|\,\underline{\theta_k}\right)$$

It is often more convenient to work with the log of the likelihood which up to an arbitrary additive constant is:

$$l(\Theta;X)=\sum_{i=1}^{n}\log\left(f_{Mixture}\left(\underline{x_i}\,|\,\underline{\theta_k}\right)\right)=\sum_{i=1}^{n}\log\left(\sum_{k=1}^{g}\pi_k f_k\left(\underline{x_i}\,|\,\underline{\theta_k}\right)\right)$$

There is generally no direct closed-form solution to maximize this log likelihood function because of the sum of terms inside the logarithm when the underlying model is a mixture distribution. Since the log-likelihood function leads to a non-linear optimization problem, many methods have been applied to solve this problem.

A popular approach, which we will now develop, is the EM (Expectation – Maximization) algorithm, first proposed by Dempster, Laird, and Rubin (1977) to handle missing data problem.

2.4 Introducing latent variables

While it is difficult to solve the maximization problem for mixture likelihood, it can be made somewhat easier by enlarging the sample with latent (unobserved) data. The latent variables are then treated as missing and the EM algorithm is applied. In the context of cluster analysis, the latent variable will be the class label of each point. The original data is then considered to be incomplete since the class labels are unknown; a complete data set would be (X, Z), where every observation has a known class label.

The mixture model is treated as a distribution in which the class labels are missing, but the class labels can be treated as random variables. If these labels were known, we could get closed-form parameters estimates in each component distribution by partitioning the data points into their respective groups.

The EM algorithm is a general iterative optimization algorithm for maximizing a likelihood function given a probabilistic model with missing data. For each EM-step the likelihood can only increase, thus guaranteeing convergence of the method to at least a local maximum of the likelihood as a function on the parameter space.

Let latent indicator vector $\underline{Z_i}$ be a g-dimensional component indicator label vector $(Z_{i1},...,Z_{ig})$ with Z_{ik} = 1, if and only if $x_i \in Group_k$; and 0, otherwise. We can easily see that $\sum_{i=1}^{g} Z_{ik} = 1$ and so $\underline{Z_i}$ is distributed according to a multinomial distribution consisting of one draw on g categories with probabilities $\pi_1,...,\pi_g$; that is:

$$\underline{Z_i} \sim f(Z_i) = Mult(1, \underline{\pi}) = \binom{1}{z_1, z_2, ..z_g} \prod_{k=1}^{g} (\pi_k)^{z_{ik}} = \prod_{k=1}^{g} (\pi_k)^{z_{ik}}$$

where $\underline{\pi} = (\pi_1,...,\pi_g)^T$. The number of observations within group k can be obtained by summing over all the indicator variables z_{ik} for all observations inside group k; that is

$n_k = \sum_{i=1}^{n} z_{ik}$ and $\sum_{i=1}^{n} n_k = n$.

Similarly, the density $f(\underline{x} \mid Z)$ is $f_k(\underline{x_i})$ when $\underline{Z_{ik}} = 1$ or simply $\prod_{k=1}^{g} [f_k(\underline{x})]^{z_{ik}}$. And $f(Z)$ is π_k when $\underline{Z_k} = 1$ or simply $\prod_{k=1}^{g} [\pi_k]^{z_{ik}}$. The joint density of (X, Z) is therefore $f(\underline{x}, Z) = f(\underline{x} \mid Z) f(Z) = \prod_{k=1}^{g} [f_k(\underline{x})]^{z_{ik}} \prod_{k=1}^{g} [\pi_k]^{z_{ik}} f(Z)$.

An observation $\underline{x_i}$ can be considered to be drawn from one of a fixed number of component distributions according to the probabilities $\pi_1, ..., \pi_g$, and then conditionally on being in group k, drawn from the density $f_k(\underline{x})$. That is the π_k is the probability of coming from group k and $f_k(\underline{x})$ is the conditional probability of x given it comes from group k.

The probability, τ_{ik}, of observation i belonging to group k given the values of $\underline{x_i}$ can now be calculated by Bayes' theorem:

$$\tau_{ik} = \Pr(x_i \in Group_k \mid x_i) = \frac{\Pr(x_i \mid x_i \in Group_k) \Pr(x_i \in Group_k)}{\Pr(x_i)} = \frac{\pi_k f_k(x_i; \underline{\theta_k})}{\sum_{k=1}^{g} \pi_k f_k(x_i; \underline{\theta_k})}$$

For a particular observed $\underline{x_i}$, we evaluate this membership probability for each group, and assign it to the group having the greatest probability. That is, if $\tau_{ik} = \max(\tau_{i1}, \tau_{i2}, ..., \tau_{ig})$, we assign observation i to group k, and so might estimate z_{ik} to be 1 and 0, otherwise.

3. EM Algorithm

3.1 EM In general

EM algorithm is developed to find maximum likelihood estimators with missing data. The log-likelihood function $l(\Theta;X)$ can be written as the difference between two likelihood functions as follows

$$\begin{aligned} l(\Theta;X) &= \log Lik(\Theta;X) \\ &= \log f(X;\Theta) \\ &= \log \frac{f(X,Z;\Theta)}{f(X,Z;\Theta)/f(X;\Theta)} \\ &= \log \frac{f(X,Z;\Theta)}{f(Z\mid X;\Theta)} \\ &= \log f(X,Z;\Theta) - \log f(Z\mid X;\Theta) \\ &= l_0(\Theta;X,Z) - l_1(\Theta;Z\mid X) \end{aligned}$$

where the first term is the complete data log-likelihood function. The second term $l_1(\Theta;Z\mid X)$ is the conditional log-likelihood function based on the latent variables Z given X.

Unfortunately, $l_0(\Theta;X,Z)$ and $l_1(\Theta;Z\mid X)$ require the value of Z, which is missing. Rather than maximize $l(\Theta;X)$ directly, we consider maximizing its expectation over Z given X, since $E_{Z|X}\left[l(\Theta;X),\Theta^*\right] = l(\Theta;X)$ for any value of Θ^*. It turns out that it will only be necessary to maximize $E_{Z|X}\left[l_0(\Theta;X,Z),\Theta^*\right]$ at any step.

Consider $l(\Theta;X)$ as a function of the dummy parameter Θ , maximizing the expectation of $l(\Theta;X)$ over $Z \mid X$ based on the current choice Θ^* gives

$$
\begin{aligned}
l(\Theta;X) &= E_{Z|X}\left[l(\Theta;X),\Theta^*\right] \\
&= E_{Z|X}\left[l_0(\Theta;X,Z),\Theta^*\right] - E_{Z|X}\left[l_1(\Theta;Z \mid X),\Theta^*\right] \\
&= Q(\Theta,\Theta^*) - R(\Theta,\Theta^*)
\end{aligned}
$$

where Θ^* is the current choice and Θ is the true value. Here $Q(\Theta,\Theta^*)$ is defined as $E_{Z|X}\left[l_0(\Theta;X,Z),\Theta^*\right]$; $R(\Theta,\Theta^*)$ is defined as $E_{Z|X}\left[l_1(\Theta;Z \mid X),\Theta^*\right]$. This unfortunately is a function of the true, say but unknown value Θ^*, which is to be estimated.

However this suggests a possible iterative procedure. Begin with an initial value for $\Theta^{(0)}$ and find the expectation for $l(\Theta;X)$ over $Z \mid X$ as if $\Theta^{(0)}$ were the true value. Maximize this expectation as a function of Θ^* to get a new value for Θ, say $\Theta^{(t)}$. Using this value for the true value of Θ perform again the expectation step followed by the maximization step. Repeat these until there is no change in $\Theta^{(t)}$. The above procedure works only when it guarantee converge.

Choosing $\Theta^* = \Theta^{(t)}$, find $\Theta^{(t+1)}$ which maximizes $Q(\Theta,\Theta^*)$, where the superscript t inside the bracket denotes step of iteration and $\Theta^{(t)}$ denotes the parameter estimate Θ at loop t.

The difference of log-likelihood between iterations can be written as

$$
l(\Theta^{(t+1)};X) - l(\Theta^{(t)};X)
$$

$$= [Q\left(\Theta^{(t+1)},\Theta^{*}\right) - R\left(\Theta^{(t+1)},\Theta^{*}\right)] - [Q\left(\Theta^{(t)},\Theta^{*}\right) - R\left(\Theta^{(t)},\Theta^{*}\right)]$$
$$= [Q\left(\Theta^{(t+1)},\Theta^{*}\right) - Q\left(\Theta^{(t)},\Theta^{*}\right)] - [R\left(\Theta^{(t+1)},\Theta^{*}\right) - R\left(\Theta^{(t)},\Theta^{*}\right)]$$

Choose Θ^{*} to be $\Theta^{(t)}$,

$$R\left(\Theta^{(t+1)},\Theta^{(t)}\right) - R\left(\Theta^{(t)},\Theta^{(t)}\right)$$
$$= E_{Z|X}\left\lfloor l\left(\Theta^{(t+1)};Z \mid X\right)\Theta^{(t)}\right\rfloor - E_{Z|X}\left\lfloor l\left(\Theta^{(t)};Z \mid X\right)\Theta^{(t)}\right\rfloor$$
$$= E_{Z|X}\left[\log\left\{\frac{f\left(Z \mid X;\Theta^{(t+1)}\right)}{f\left(Z \mid X;\Theta^{(t)}\right)}\right\};\Theta^{(t)}\right]$$
$$<= \log\left[E_{Z|X}\left\{\frac{f\left(Z \mid X;\Theta^{(t+1)}\right)}{f\left(Z \mid X;\Theta^{(t)}\right)}\right\};\Theta^{(t)}\right]$$
$$= \log\left[\int_{Z}\frac{f\left(Z \mid X;\Theta^{(t+1)}\right)}{f\left(Z \mid X;\Theta^{(t)}\right)}f\left(Z \mid X;\Theta^{(t)}\right)dz\right]$$
$$= \log\int_{Z} f\left(Z \mid X;\Theta^{(t+1)}\right)dz$$
$$= \log 1$$
$$= 0$$

where the inequality follows from Jensen's inequality that $E[f(x)] <= f[E(x)]$ if $f(x)$ is a convex function. Since the logarithm transformation is a convex function, it follows then that minus this difference $R\left(\Theta^{(t+1)},\Theta^{(t)}\right) - R\left(\Theta^{(t)},\Theta^{(t)}\right)$ is a non-negative value. Now the M-step of the EM algorithm is to choose $\Theta^{(t+1)}$ so that $Q\left(\Theta^{(t+1)},\Theta^{(t)}\right) >= Q\left(\Theta^{(t)},\Theta^{(t)}\right)$ for any Θ, including $\Theta = \Theta^{(t)}$. That means the difference $l\left(\Theta^{(t+1)};X\right) - l\left(\Theta^{(t)};X\right)$ will be non-negative.

This guarantees that the EM iteration never decreases the log-likelihood, and will converge to a maximum (local or global) finally. EM algorithm maximizes $l(\Theta; X)$ through maximizing $Q(\Theta, \Theta^{(t)})$ over parameter space of Θ. That means EM algorithm will look at the complete data log-likelihood only.

EM algorithm can be summarized by iteratively executing E-step and M-step, which starts with initial value of $\Theta^{*} = \Theta^{(0)}$. The E-step is to find the conditional expectation of the latent variable Z estimated conditionally on the current parameter from last step. Instead of find the log-likelihood of data X, The M-step is the procedure of maximizing this conditional expectation of the complete data log-likelihood function $Q(\Theta, \Theta^{(t)})$ over all Θ (i.e. $Q(\Theta^{(t+1)}, \Theta^{(t)}) >= Q(\Theta, \Theta^{(t)})$ for all Θ). So as to get the next step parameter $\Theta^{(t+1)}$ at the loop (t+1). Repeat these until there is no change in $\Theta^{(t)}$.

3.2 EM for finite MVN mixture

3.2.1 Estimation in general

If the g component density functions are taken to be multivariate normal, the kth component multivariate normal density function, with mean vector $\underline{u_k}$ and covariance matrix $\sum_k$, is written as

$$f_k(\underline{x_i}) = \phi(\underline{x_i} \mid \underline{u_k}, \textstyle\sum_k) = (2\pi)^{-\frac{p}{2}} \left|\textstyle\sum_k\right|^{-1/2} \exp\left\{-\frac{1}{2}(\underline{x_i} - \underline{u_k})^T \textstyle\sum^{-1}(\underline{x_i} - \underline{u_k})\right\}$$

The finite mixture model becomes a finite multivariate normal mixture with the form

$$f_{Mixture}(\underline{x_i} \mid \Theta) = \sum_{k=1}^{g} \pi_k \phi(\underline{x_i} \mid \underline{u_k}, \textstyle\sum_k)$$

where the parameter Θ is written as $\{\pi_1,\ldots,\pi_{g-1},\underline{u_1},\ldots,\underline{u_g},\Sigma_1,\ldots,\Sigma_g\}$

The likelihood of complete data can be written as:

$$\begin{aligned}
L(\Theta;X,Z) &\propto \prod_{i=1}^{n} f\left(X_i,\underline{Z_i}\right) \\
&= \prod_{i=1}^{n} f\left(X_i \mid \underline{Z_i}\right) f\left(\underline{Z_i}\right) \\
&= \prod_{i=1}^{n}\left\{\left(\prod_{k=1}^{g}\left[\phi\left(x_i;\underline{u_k},\Sigma_k\right)\right]^{Zik}\right)\left(\prod_{k=1}^{g}\left(\pi_k\right)^{Zik}\right)\right\} \\
&= \prod_{i=1}^{n}\prod_{k=1}^{g}\left[\phi\left(x_i;\underline{u_k},\Sigma_k\right)\right]^{Zik}\left(\pi_k\right)^{Zik}
\end{aligned}$$

and the log-likelihood (up to an additive constant) is

$$l(\Theta;X,Z) = \sum_{i=1}^{n}\sum_{k=1}^{g} Z_{ik}\left\{\log\pi_k + \log\left(\phi\left(\underline{x_i};\underline{u_k},\Sigma_k\right)\right)\right\}$$

The EM algorithm simplified by introducing latent variable

$$\begin{aligned}
z_{ik} &= 1, \text{ if and only if } x_i \in Group_k; \\
&= 0, \text{ otherwise}
\end{aligned}$$

Note that conditional on X, is a Bernoulli random variable with probability τ_{ik} for $z_{ik} = 1$, therefore $E_{Z|X}\left(z_{ik}\right) = 1 \times \Pr(z_{ik} = 1) + 0 \times \Pr(z_{ik} = 0) = \tau_{ik}$

Hence, we get the conditional expectation of the log-likelihood as follows

$$E_{Z|X}\, l(\Theta;X,Z) = \sum_{i=1}^{n}\sum_{k=1}^{g} E_{Z|X}\left(Z_{ik}\right)\left\{\log\pi_k + \log\left(\phi\left(\underline{x_i};\underline{u_k},\Sigma_k\right)\right)\right\}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{g} \tau_{ik} \left\{ \log \pi_k + \log\left(\phi\left(\underline{x_i}; \underline{u_k}, \Sigma_k\right)\right) \right\}$$

where τ_{ik} is the probability of observation i belonging to group k

The expectation of $l(\Theta; X, Z)$ over $Z \mid X$ based on current parameter choice Θ^* is

$$Q(\Theta, \Theta^*)$$

$$= E_{Z|X}\left[l(\Theta; X, Z) \mid \Theta^* \right]$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{g} E_{Z|X}\left[Z_{ik} \mid x_i; \Theta^* \right] \left\{ \log \pi_k + \log\left(\phi\left(\underline{x_i}; \underline{u_k}, \Sigma_k\right)\right) \right\}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{g} \tau_{ik}^{\ *} \left\{ \log \pi_k + \log\left(\phi\left(\underline{x_i}; \underline{u_k}, \Sigma_k\right)\right) \right\}$$

$$= \sum_{i=1}^{n} \sum_{k=1}^{g} \tau_{ik}^{\ *} \left\{ \log \pi_k \right\} + \sum_{i=1}^{n} \sum_{k=1}^{g} \tau_{ik}^{\ *} \log\left(\phi\left(\underline{x_i}; \underline{u_k}, \Sigma_k\right)\right)$$

$$= \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \left\{ \log \pi_k \right\} + \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \log\left(\phi\left(\underline{x_i}; \underline{u_k}, \Sigma_k\right)\right)$$

$$= \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \left\{ \log \pi_k \right\} + \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \log\left((2\pi)^{-\frac{p}{2}} \mid \Sigma_k \mid^{-1/2} \exp\left\{ -\tfrac{1}{2} \left(\underline{x_i} - \underline{u_k}\right)^T \Sigma_k^{-1} \left(\underline{x_i} - \underline{u_k}\right) \right\} \right)$$

$$= \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \left\{ \log \pi_k \right\} + \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \log(2\pi)^{-\frac{p}{2}} + \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \log \mid \Sigma_k \mid^{-1/2} + \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \left\{ -\tfrac{1}{2} \left(\underline{x_i} - \underline{u_k}\right)^T \Sigma_k^{-1} \left(\underline{x_i} - \underline{u_k}\right) \right\}$$

$$= \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \left\{ \log \pi_k \right\} + \frac{np}{2} \log(2\pi) + \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \log \mid \Sigma_k \mid^{-1/2} + \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{\ *} \left\{ -\tfrac{1}{2} \left(\underline{x_i} - \underline{u_k}\right)^T \Sigma_k^{-1} \left(\underline{x_i} - \underline{u_k}\right) \right\}$$

where $\tau_{ik}^{\ *}$ is the probability of observation i belonging to group k based on the current parameter choice Θ^*, and can be calculated by

$$E_{Z|X}\left(z_{ik} \mid x_i; \pi_k^*, \underline{u}_k^*, \Sigma_k^* \right) \overbrace{=}^{Bayes} \frac{\pi_k^* \phi\left(\underline{x_i} \mid \underline{u}_k^*, \Sigma_k^*\right)}{\sum_{k=1}^{g} \pi_k^* \phi\left(\underline{x_i} \mid \underline{u}_k^*, \Sigma_k^*\right)}$$

Find the estimated $\widehat{\Theta}$, which maximizes $Q(\Theta,\Theta^*)$ for fixed Θ^*, is the same as to find the parameter vector $\left(\widehat{\pi}_1,...,\widehat{\pi}_k;\widehat{u}_1,..,\widehat{u}_k;\widehat{\sum}_1,..,\widehat{\sum}_k\right)$ which maximizes

$$Q(\Theta,\Theta^*)=Q\left(\pi_1,...,\pi_k,u_1,..,u_k,\Sigma_1,..,\Sigma_k,\pi_1^*,...,\pi_k^*,u_1^*,..,u_k^*,\Sigma_1^*,..,\Sigma_k^*\right)$$

with subject to the equation $\sum_{k=1}^{g}\pi_k=1$.

Using a Lagrange multiplier, $\max_x f(x)$ subject to $g(x)=c \Leftrightarrow \max_x\{f(x)+\lambda(c-g(x))\}$, we know that if we set $c=1$, the above maximization problem is equivalent to find $\left(\widehat{\pi}_1,...,\widehat{\pi}_k;\widehat{u}_1,..,\widehat{u}_k;\widehat{\sum}_1,..,\widehat{\sum}_k,\widehat{\lambda}\right)$, which maximizes

$$Q\left(\pi_1,...,\pi_k,u_1,..,u_k,\Sigma_1,..,\Sigma_k,\pi_1^*,...,\pi_k^*,u_1^*,..,u_k^*,\Sigma_1^*,..,\Sigma_k^*\right)+\lambda\left(1-\sum_{k=1}^{g}\pi_k\right).$$

The maximum value for parameters is found by setting the first derivative of function $Q(\Theta,\Theta^*)$ or $Q(\Theta,\Theta^*)+\lambda\left(1-\sum_{k=1}^{g}\pi_k\right)$ as 0. Since the log function is always monotonic increasing, the second derivative is not need to be checked. The detail of the maximization procedure are given as follows:

Maximize function $Q(\Theta,\Theta^*)+\lambda\left(1-\sum_{k=1}^{g}\pi_k\right)$ with respect to π_j

$$\frac{\partial}{\partial\pi_j}\left\{Q(\Theta,\Theta^*)+\lambda\left(1-\sum_{k=1}^{g}\pi_k\right)\right\}$$

$$=\frac{\partial}{\partial\pi_j}\{\sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}^*\{\log\pi_k\}+\frac{np}{2}\log(2\pi)$$

$$+\sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}^*\log|\Sigma_k|^{-1/2}+\sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}^*\left\{\tfrac{1}{2}\left(\underline{x_i}-\underline{u_k}\right)^T\Sigma_k^{-1}\left(\underline{x_i}-\underline{u_k}\right)\right\}+\lambda\left(1-\sum_{k=1}^{g}\pi_k\right)\}$$

$$= \frac{\partial}{\partial \pi_j}\{\sum_{k=1}^{g}\{\log \pi_k\}\sum_{i=1}^{n}\tau_{ik}^{\ *} + \lambda\left(1-\sum_{k=1}^{g}\pi_k\right)\}$$

$$= \frac{\partial}{\partial \pi_j}\{\log \pi_k \sum_{i=1}^{n}\tau_{ij}^{\ *} + \sum_{k \neq j}^{g}\{\log \pi_k\}\sum_{i=1}^{n}\tau_{ik}^{\ *} + \lambda(-\pi_k) + \lambda\left(1-\sum_{k \neq j}^{g}\pi_k\right)\}$$

$$= \frac{1}{\pi_j}\sum_{i=1}^{n}\tau_{ij}^{\ *} - \lambda$$

Set $\frac{\partial}{\partial \pi_j}\left\{Q\left(\Theta,\Theta^*\right) + \lambda\left(1-\sum_{k=1}^{g}\pi_k\right)\right\} = 0$, we get $\hat{\lambda} = \frac{1}{\hat{\pi}_j}\sum_{i=1}^{n}\tau_{ij}^{\ *}$ and $\hat{\pi}_j = \frac{\sum_{i=1}^{n}\tau_{ij}^{\ *}}{\hat{\lambda}}$

Since $1-\sum_{k=1}^{g}\hat{\pi}_k = 1-\sum_{k=1}^{g}\frac{\sum_{i=1}^{n}\tau_{ij}^{\ *}}{\hat{\lambda}} = 1-\frac{1}{\hat{\lambda}}\sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ij}^{\ *} = 1-\frac{1}{\hat{\lambda}}\sum_{k=1}^{g}(1) = 1-\frac{n}{\hat{\lambda}} = 0$, we get $\hat{\lambda} = n$.

So the estimated mixture component proportion $\hat{\pi}_j = \frac{\sum_{i=1}^{n}\tau_{ij}^{\ *}}{\hat{\lambda}} = \frac{\sum_{i=1}^{n}\tau_{ij}^{\ *}}{n}$

Maximize function $Q\left(\Theta,\Theta^*\right)$ with respect to $\underline{u_j}$

$$= \frac{\partial}{\partial \underline{u_j}}\{\sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}^{\ *}\{\log \pi_k\} + \frac{np}{2}\log(2\pi)$$

$$+ \sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}^{\ *}\log|\Sigma_k|^{-1/2} + \sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}^{\ *}\left\{\tfrac{1}{2}\left(\underline{x_i}-\underline{u_k}\right)^T\Sigma_k^{\ -1}\left(\underline{x_i}-\underline{u_k}\right)\right\}$$

$$= \frac{\partial}{\partial \underline{u_j}}\sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}^{\ *}\left\{\tfrac{1}{2}\left(\underline{x_i}-\underline{u_k}\right)^T\Sigma_k^{\ -1}\left(\underline{x_i}-\underline{u_k}\right)\right\}$$

$$= \frac{\partial}{\partial \underline{u_j}}\sum_{i=1}^{n}\left(-\frac{\tau_{ij}^{\ *}}{2}\right)\left\{\left(\underline{x_i}-\underline{u_j}\right)^T\Sigma_j^{\ -1}\left(\underline{x_i}-\underline{u_j}\right)\right\}$$

$$= \frac{\partial}{\partial \underline{u}_j} \sum_{i=1}^{n} \left(- \frac{\tau_{ij}^{*}}{2} \right) \left\{ \underline{x}_i^{T} \Sigma_j^{-1} \underline{x}_i - \underline{u}_i^{T} \Sigma_j^{-1} \underline{x}_i - \underline{x}_i^{T} \Sigma_j^{-1} \underline{u}_j + \underline{u}_j^{T} \Sigma_j^{-1} \underline{u}_j \right\}$$

$$= \frac{\partial}{\partial \underline{u}_j} \sum_{i=1}^{n} \left(- \frac{\tau_{ij}^{*}}{2} \right) \left\{ \underline{x}_i^{T} \Sigma_j^{-1} \underline{x}_i - 2\underline{x}_i^{T} \Sigma_j^{-1} \underline{u}_j + \underline{u}_j^{T} \Sigma_j^{-1} \underline{u}_j \right\}$$

$$= \sum_{i=1}^{n} \left(- \frac{\tau_{ij}^{*}}{2} \right) \left\{ 2\Sigma_j^{-1} \underline{x}_i + 2\Sigma_j^{-1} \underline{u}_j \right\}$$

$$= \left(\Sigma_j^{-1} \right) \sum_{i=1}^{n} \left(\tau_{ij}^{*} \right) \left\{ \underline{u}_j - \underline{x}_i \right\}$$

$$= \left(\Sigma_j^{-1} \right) \left(\sum_{i=1}^{n} \tau_{ij}^{*} \underline{u}_j - \sum_{i=1}^{n} \tau_{ij}^{*} \underline{x}_i \right)$$

Set $\frac{\partial Q(\Theta, \Theta^{*})}{\partial \underline{u}_j} = 0$, and Σ_j^{-1} is invertible, we get $\sum_{i=1}^{n} \tau_{ij}^{*} \underline{u}_j - \sum_{i=1}^{n} \tau_{ij}^{*} \underline{x}_i = 0$. So the estimated normal component mean vector $\underline{\widehat{u}}_j = \frac{\sum_{i=1}^{n} \tau_{ij}^{*} \underline{x}_i}{\sum_{i=1}^{n} \tau_{ij}^{*}}$

To maximize function $Q(\Theta, \Theta^{*})$ with respect to Σ_j, we write

$$Q(\Theta, \Theta^{*}) = \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{*} \{ \log \pi_k \} + \frac{np}{2} \log(2\pi)$$

$$+ \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{*} \log |\Sigma_k|^{-1/2} + \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{*} \left\{ \tfrac{1}{2} \left(\underline{x}_i - \underline{u}_k \right)^{T} \Sigma_k^{-1} \left(\underline{x}_i - \underline{u}_k \right) \right\}$$

(The first two terms don’t depend on Σ_j, and can be dropped in the maximization problem)

$$= \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{*} \log |\Sigma_k|^{-1/2} + \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}^{*} \left\{ \tfrac{1}{2} \left(\underline{x}_i - \underline{u}_k \right)^{T} \Sigma_k^{-1} \left(\underline{x}_i - \underline{u}_k \right) \right\}$$

(Any term not including Σ_j can also be dropped)

$$= \sum_{i=1}^{n} \tau_{ij}^{*} \log|\Sigma_j|^{-1/2} + \sum_{i=1}^{n} \tau_{ij}^{*} \left\{ -\tfrac{1}{2} \left(\underline{x_i} - \underline{u_j}\right)^T \Sigma_j^{-1} \left(\underline{x_i} - \underline{u_j}\right) \right]$$

$$= \left(-\tfrac{1}{2}\right) \sum_{i=1}^{n} \tau_{ij}^{*} \log|\Sigma_j| - \left(\tfrac{1}{2}\right) \sum_{i=1}^{n} \tau_{ij}^{*} tr \left\{ \left(\underline{x_i} - \underline{u_j}\right)^T \Sigma_j^{-1} \left(\underline{x_i} - \underline{u_j}\right) \right]$$

$$= \left(-\tfrac{1}{2}\right) \sum_{i=1}^{n} \tau_{ij}^{*} \log|\Sigma_j| - \left(\tfrac{1}{2}\right) \sum_{i=1}^{n} \tau_{ij}^{*} tr \left\{ \Sigma_j^{-1} \left(\underline{x_i} - \underline{u_j}\right)\left(\underline{x_i} - \underline{u_j}\right)^T \right]$$

$$= \left(-\tfrac{1}{2}\right) \sum_{i=1}^{n} \tau_{ij}^{*} \log|\Sigma_j| - \left(\tfrac{1}{2}\right) tr \left(\sum_{i=1}^{n} \tau_{ij}^{*} \left\{ \Sigma_j^{-1} \left(\underline{x_i} - \underline{u_j}\right)\left(\underline{x_i} - \underline{u_j}\right)^T \right\} \right)$$

$$= \left(-\tfrac{1}{2}\right) \sum_{i=1}^{n} \tau_{ij}^{*} \log|\Sigma_j| - \left(\tfrac{1}{2}\right) tr \left(\Sigma_j^{-1} \sum_{i=1}^{n} \tau_{ij}^{*} \left\{ \left(\underline{x_i} - \underline{u_j}\right)\left(\underline{x_i} - \underline{u_j}\right)^T \right\} \right)$$

So maximizing $Q\left(\Theta, \Theta^{*}\right)$ with respect to Σ_j is equivalent to maximizing the above expression with respect to Σ_j. Need to find $\hat{\Sigma}_j$, which maximizes

$$\left(-\tfrac{1}{2}\right) \sum_{i=1}^{n} \tau_{ij}^{*} \log|\Sigma_j| - \left(\tfrac{1}{2}\right) tr \left(\Sigma_j^{-1} \sum_{i=1}^{n} \tau_{ij}^{*} \left\{ \left(\underline{x_i} - \underline{u_j}\right)\left(\underline{x_i} - \underline{u_j}\right)^T \right\} \right)$$

Theorem 1: For any $p \times p$ p. d. matrix S, and positive constants a and b,

$$|\Sigma|^{-b} \exp\left\{ -a\left[tr\left(\Sigma^{-1} S\right)\right]\right\} \le |aS/b|^{-b} \exp\{-pb\}$$

or

$$-b\log|\Sigma| - a\left[tr\left(\Sigma^{-1} S\right)\right] \le -b\log|aS/b| - pb$$

or

$$b\log|\Sigma| + a\left[tr\left(\Sigma^{-1} S\right)\right] \ge b\log|aS/b| + pb$$

for all $p \times p$ p. d. matrices Σ with equality holds if and only if $\Sigma = aS/b$.

Proof: See Srivastava and Khatri (page25, 1979)

So if we set $a = \frac{1}{2}$, $b = \frac{1}{2}\sum_{i=1}^{n}\tau_{ij}^{*} > 0$, and $S = \sum_{i=1}^{n}\tau_{ij}^{*}\left(\underline{x}_i - \underline{\hat{u}}_j\right)\left(\underline{x}_i - \underline{\hat{u}}_j\right)^T$. Notice that S is a positive definite matrix because $V = \sum_{i=1}^{n}\left(\underline{x}_i - \underline{a}\right)\left(\underline{x}_i - \underline{a}\right)^T$ is positive definite matrix for any vector $\underline{a}$. That means the points $\underline{x}_1, \underline{x}_2, ..., \underline{x}_n$ do not lie in a lower dimensional linear space (manifold) and $\underline{y}^T V \underline{y} = \sum_{i=1}^{n}\left(\underline{y}^T\left(\underline{x}_i - \underline{a}\right)\right)^2 > 0$. Then we know at least one $\left(\underline{y}^T\left(\underline{x}_i - \underline{a}\right)\right)^2 > 0$. So for positive constants $w_i = \tau_{ij}$ (for all i), that is $\underline{y}^T S \underline{y} = \sum_{i=1}^{n} w_i\left(\underline{y}^T\left(\underline{x}_i - \underline{a}\right)\right)^2 > 0$, then $S = \sum_{i=1}^{n} w_i\left(\underline{x}_i - \underline{a}\right)\left(\underline{x}_i - \underline{a}\right)^T$ will be positive definite. (In the case of V do not lie in a lower dimensional linear space, we can just first move all data to the lower dimensional manifold.)

Then the term

$$\left[\left(-\tfrac{1}{2}\right)\sum_{i=1}^{n}\tau_{ij}^{*}\log|\Sigma_j| - \left(\tfrac{1}{2}\right)tr\left(\Sigma_j^{-1}\sum_{i=1}^{n}\tau_{ij}^{*}\left\{\left(\underline{x}_i - \underline{u}_j\right)\left(\underline{x}_i - \underline{u}_j\right)^T\right\}\right)\right]$$

will be bounded by a maximum value, and will get the maximum value if and only if

$$\Sigma_j = aS/b = \frac{\frac{1}{2}S}{\frac{1}{2}\sum_{i=1}^{n}\tau_{ij}^{*}} = \frac{\sum_{i=1}^{n}\tau_{ij}^{*}\left(\underline{x}_i - \underline{\hat{u}}_j\right)\left(\underline{x}_i - \underline{\hat{u}}_j\right)^T}{\sum_{i=1}^{n}\tau_{ij}^{*}},$$

which is $\hat{\Sigma}_j$, the maximum likelihood estimate of Σ_j.

For the purpose of computational efficiency, the final component parameter estimates can be simplified by introducing the statistics

$$T_{k1}^{(t)} = \sum_{i=1}^{n} \tau_{ik}^{(t)} \text{ , } T_{k2}^{(t)} = \sum_{i=1}^{n} \tau_{ik}^{(t)} x_i \text{ and } T_{k3}^{(t)} = \sum_{i=1}^{n} \tau_{ik}^{(t)} x_i x_i^{T} .$$

The kth component proportion estimates can be re-written as

$$\underline{\hat{\pi}_k} = T_{k1} / n$$

The kth component mean vector estimates can be re-written as

$$\underline{\hat{u}_k} = T_{k2} / T_{k1}$$

The kth component covariance matrix estimates can be re-written as

$$\hat{\Sigma}_k = \left\{ T_{k3} - T_{k1}^{-1} T_{k2} T_{k2}^{T} \right\} / T_{k1} ,$$

since

$$\hat{\Sigma}_k = \frac{\sum_{i=1}^{n} \tau_{ik}{}^{*} \left(\underline{x_i} - \underline{\hat{u}_k}\right)\left(\underline{x_i} - \underline{\hat{u}_k}\right)^{T}}{\sum_{i=1}^{n} \tau_{ik}{}^{*}}$$

$$= \frac{\sum_{i=1}^{n} \tau_{ik}{}^{*} \left(\underline{x_i x_i}^{T} - \underline{\hat{u}_k x_i}^{T} - \underline{x_i \hat{u}_k}^{T} + \underline{\hat{u}_k \hat{u}_k}^{T}\right)}{\sum_{i=1}^{n} \tau_{ik}{}^{*}}$$

$$= \frac{\sum_{i=1}^{n} \tau_{ik}{}^{*} \underline{x_i x_i}^{T} - \underline{\hat{u}_k}\left(\sum_{i=1}^{n} \tau_{ik}{}^{*} \underline{x_i}^{T}\right) - \left(\sum_{i=1}^{n} \tau_{ik}{}^{*} \underline{x_i}\right)\underline{\hat{u}_k}^{T} + \left(\sum_{i=1}^{n} \tau_{ik}{}^{*}\right)\underline{\hat{u}_k \hat{u}_k}^{T}}{\sum_{i=1}^{n} \tau_{ik}{}^{*}}$$

$$= \frac{T_{k3} - \underline{\hat{u}_k} T_{k2}^{T} - T_{k2} \hat{u}_k^{T} + T_{k1} \hat{u}_k \hat{u}_k^{T}}{T_{k1}}$$

$$= \frac{T_{k3} - \frac{T_{k2}}{T_{k1}} T_{k2}^{T} - T_{k2}\left(\frac{T_{k2}}{T_{k1}}\right)^{T} + T_{k1} \frac{T_{k2}}{T_{k1}}\left(\frac{T_{k2}}{T_{k1}}\right)^{T}}{T_{k1}}$$

$$= \frac{T_{k3} - \left(\frac{1}{T_{k1}} + \frac{1}{T_{k1}^T} - \frac{1}{T_{k1}^T} \right) T_{k2} T_{k2}^T}{T_{k1}}$$

$$= \left\{ T_{k3} - T_{k1}^{-1} T_{k2} T_{k2}^T \right\} / T_{k1}$$

3.2.1.1Some useful consequences of Theorem 1

Corollary 1: The $p \times p$ symmetric matrix M such that $|M| = 1$ minimizing $tr\left(QM^{-1}\right)$ where Q is a symmetric positive definite matrix is

$$M = \frac{Q}{|Q|^{1/p}},$$

and the minimized value is $p|Q|^{1/p}$.

Proof: Consider finding the $p \times p$ p. d. matrix Σ, which minimizing $\log|\Sigma| + tr\left(\Sigma^{-1} S\right)$, for any $p \times p$ p. d. matrix S. This is a special case of Theorem 1, when a=b=1. Now let, without lost of generality, $\Sigma = \alpha M$, where $|M| = 1$ and $\alpha > 0$. Then we consider finding α and M, which minimizes

$$\log|\alpha M| + tr\left(\alpha^{-1} M^{-1} S\right) = p\log(\alpha) + \log|M| + \alpha^{-1} tr\left(M^{-1} S\right)$$

$$= p\log(\alpha) + \log|M| + \alpha^{-1} tr\left(M^{-1} S\right)$$

$$= p\log(\alpha) + \alpha^{-1} tr\left(M^{-1} S\right)$$

Minimizing first with respect to M is equivalent to minimizing $tr\left(M^{-1} S\right)$. So to find M, which minimizes $tr\left(M^{-1} S\right)$, we need only minimizing $p\log(\alpha) + \alpha^{-1} tr\left(M^{-1} S\right)$ with respect to M and α. From Theorem 1, $p\log(\alpha) + \alpha^{-1} tr\left(M^{-1} S\right)$ is minimized with respect to Σ when $\Sigma = S$ or when $\alpha M = S$ or $M = \alpha^{-1} S$. Since $|M| = 1$, $\alpha = |S|^{1/p}$ and $M = \frac{S}{|S|^{1/p}}$. The minimum value achieved is

$$tr\left(M^{-1}S\right) = tr\left[\left(\frac{S}{|S|^{1/p}}\right)^{-1} S\right] = tr\left(|S|^{1/p} S^{-1} S\right) = tr\left(|S|^{1/p}\right)$$

Corollary 2: The $p \times p$ diagonal matrix M minimizing $tr\left(QM^{-1}\right) + \alpha \log|M|$ where Q is a symmetric p. d. matrix and α is a positive real number is $M = \left(1/\alpha\right) diag(Q)$.

Proof: This is a special case of Theorem 1, when $\frac{b}{a} = \alpha$, $\Sigma = M$ and $S = Q$. $tr\left(QM^{-1}\right) + \alpha \log|M|$ get its minimum when $M = \frac{a}{b} Q = \frac{1}{\alpha} Q$.

Suppose M is a diagonal matrix, then M^{-1} is also a diagonal matrix. Since

$$\begin{aligned} tr\left(QM^{-1}\right) + \alpha \log|M| &= tr\left(M^{-1}Q\right) + \alpha \log|M| \\ &= tr\left(M^{-1} diag(Q)\right) + \alpha \log|M| \end{aligned}$$

Find M, which minimizes $tr\left(QM^{-1}\right) + \alpha \log|M|$, is $\hat{M} = \left(1/\alpha\right) diag(Q)$

Theorem 2: The orthogonal matrix Q, minimizing $tr\left(QAQ^{-1}B\right)$ where A and B are diagonal matrices, with general diagonal term α_j and β_j such that $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_p$ and $\beta_1 \leq \beta_2 \leq \ldots \leq \beta_p$, is the identity matrix and the minimized value is $tr(AB) = \sum_{j=1}^{p} \alpha_j \beta_j$

Proof: See theorem 1 of Celeux and Govaert (1994)

3.2.2 Estimation via geometric decomposition

When the covariance matrix is non-singular, Banfield and Raftery consider a variant of the standard spectral decomposition of a covariance matrix, namely:

$$\sum_k = \lambda_k O_k D_k O_k^T$$

where O_k is the matrix of eigenvectors of $\sum_k$ and with $|D_k| = 1$.

Since $\det(\sum_k) = \det(\lambda_k O_k D_k O_k^T) = \lambda_k{}^p \det(O_k D_k O_k^T) = \lambda_k{}^p$, we know $\lambda_k = |\sum_k|^{1/p}$.

Recall the conditional expectation of the log-likelihood as follows

$$Q(\Theta,\Theta^*) = E_{Z|X} l(\Theta; X, Z)$$

$$= \sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}{}^*\{\log\pi_k\} + \frac{np}{2}\log(2\pi)$$

$$+ \sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}{}^*\log|\Sigma_k|^{-1/2} + \sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}{}^*\left\{\tfrac{1}{2}(\underline{x_i} - \underline{u_k})^T \Sigma_k{}^{-1}(\underline{x_i} - \underline{u_k})\right\}$$

Already known maximize function $Q(\Theta,\Theta^*)$ with respect to Σ is equivalent to maximize

$$-\tfrac{1}{2}\left\{\sum_{k=1}^{g} tr(\Sigma_k{}^{-1}W_k) + \sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}\log|\sum_k|\right\},$$

or minimize the function

$$F(\Sigma \mid x_1,\ldots,x_n,\tau^*) = \sum_{k=1}^{g} tr(\Sigma_k{}^{-1}W_k) + \sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}\log|\sum_k|$$

where: $W_k = \sum_{i=1}^{n} \tau_{ik} (\underline{x_i} - \underline{u_k})(\underline{x_i} - \underline{u_k})^T$ and $tr(W_k) = \sum_{i=1}^{n} \tau_{ik} \left\| \underline{x_i} - \underline{u_k} \right\|^2$

Eight different covariance structures were considered by Banfield and Raftery. Refer back to section 2.2, each one results in a different value of $F(\Sigma \mid x_1, ..., x_n, \tau^*)$

3.2.2.1 Structure $\Sigma_k = \lambda I$

This is the simplest structure where every covariance has spherical shape and equal volume.

$$F(\Sigma \mid x_1, ..., x_n, \tau^*)$$

$$= F_1(\lambda \mid x_1, ..., x_n, \tau^*)$$

$$= \sum_{k=1}^{g} tr\left((\lambda I)^{-1} W_k\right) + \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik} \log|\lambda I|$$

$$= \lambda^{-1} tr\left(\sum_{k=1}^{g} W_k\right) + p \log(\lambda) \sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik}$$

$$= \lambda^{-1} tr(W) + np \log(\lambda)$$

where $W = \sum_{k=1}^{g} W_k = \sum_{k=1}^{g} \sum_{i=1}^{n} \left\{ \tau_{ik} (\underline{x_i} - \underline{u_k})(\underline{x_i} - \underline{u_k})^T \right\}$ and note that $\sum_{k=1}^{g} \sum_{i=1}^{n} \tau_{ik} = n$

Minimizing F_1 with respect to λ, we get $\hat{\lambda} = \frac{tr(W)}{np}$

3.2.2.2 Structure $\Sigma_k = \lambda_k I$

This is the second simplest structure where every covariance has spherical shape and different volume.

$$F(\Sigma \mid x_1,...,x_n,\tau^*)$$

$$= F_2(\lambda_1,...,\lambda_g \mid x_1,...,x_n,\tau^*)$$

$$= \sum_{k=1}^{g} tr\left((\lambda_k I)^{-1} W_k\right) + \sum_{k=1}^{g}\sum_{i=1}^{n} \tau_{ik} \log|\lambda_k I|$$

$$= \sum_{k=1}^{g} \lambda_k^{-1} tr(W_k) + p\sum_{k=1}^{g} \log(\lambda_k)\sum_{i=1}^{n}\tau_{ik}$$

Minimizing F_2 with respect to λ_j is equivalent to minimizing

$$\lambda_j^{-1} tr(W_j) + p\log(\lambda_j)\sum_{i=1}^{n}\tau_{ij}$$

Minimizing F_2 with respect to λ_j, we get

$$\widehat{\lambda}_j = \frac{tr(W_j)}{p\tau_j}, \text{ for j=1,…,g.}$$

where $\tau_j = \sum_{i=1}^{n}\tau_{ij}$

3.2.2.3 Structure $\Sigma_k = \Sigma = \lambda ODO^T$

This is the structure where all covariance are the same without any constrict about shape and orientation.

$$F(\Sigma \mid x_1,...,x_n,\tau^*)$$

$$= F_3(\Sigma \mid x_1,...,x_n,\tau^*)$$

$$= \sum_{k=1}^{g} tr\left(\Sigma^{-1} W_k\right) + \sum_{k=1}^{g}\sum_{i=1}^{n} \tau_{ik} \log|\Sigma|$$

$$= tr\left(W\Sigma^{-1}\right) + n\log|\Sigma|,$$

where W is as before. This quantity has the structure of Theorem 1. Let $b = n$, $a = 1$, and $S = W$. So the minimum is achieved if and only if

$$\widehat{\Sigma} = \frac{W}{n}$$

3.2.2.4 Structure $\Sigma_k = \lambda_k C$, where $C = ODO^T$, $\det(D) = 1$

This is the structure where every covariance has the same shape and orientation, but different volume.

$F(\Sigma \mid x_1,\ldots,x_n,\tau^*)$

$= F_4(\lambda_1,\ldots,\lambda_g, C \mid x_1,\ldots,x_n,\tau^*)$

$= \sum_{k=1}^{g} tr\left((\lambda_k C)^{-1} W_k\right) + \sum_{k=1}^{g}\sum_{i=1}^{n} \tau_{ik} \log|\lambda_k C|$

(Recall that $\det(D) = 1$, so $|\lambda_k C| = \lambda_k{}^p \det(C) = \lambda_k{}^p \det(ODO^T) = \lambda_k{}^p \det(D) = \lambda_k{}^p$)

$= \sum_{k=1}^{g} \lambda_k{}^{-1} tr\left(W_k C^{-1}\right) + p\sum_{k=1}^{g}\sum_{i=1}^{n} \tau_{ik} \log(\lambda_k)$

The minimization of F_4 has to be performed iteratively.

1. As the matrix C is kept fixed, the λ_k's minimizing F_4 is equivalent to minimizing

$$\lambda_k{}^{-1} tr\left(W_k C^{-1}\right) + p\sum_{i=1}^{n} \tau_{ik} \log(\lambda_k)$$

By setting the first derivative to 0, we get the estimate:

$$\widehat{\lambda_k} = \frac{tr\left(W_k C^{-1}\right)}{p\sum_{i=1}^{n} \tau_{ik}} = \frac{tr\left(W_k C^{-1}\right)}{p\tau_k}$$

2. As the volumes λ_k's are kept fixed, the matrix C minimizing F_4 is minimizing

$$tr\left(\sum_{k=1}^{g}\left(\frac{1}{\lambda_k}\right)W_k\right)C^{-1}$$

By Corollary 1, let $Q = \sum_{k=1}^{g}\left(\frac{1}{\lambda_k}\right)W_k$ and $M = C$, the function F_4 is minimized when

$$\widehat{C} = \frac{\sum_{k=1}^{g}\left(\frac{1}{\lambda_k}\right)W_k}{\left|\sum_{k=1}^{g}\left(\frac{1}{\lambda_k}\right)W_k\right|^{1/p}}$$

3.2.2.5 Structure $\Sigma_k = \lambda O_k D O_k{}^T$

This is the structure where every covariance has the same shape and volume, but with different orientation.

$$F(\Sigma \mid x_1,\ldots,x_n,\tau^*)$$

$$= F_5(\lambda, O_1,\ldots,O_g, D \mid x_1,\ldots,x_n,\tau^*)$$

$$= \sum_{k=1}^{g} tr\left(\left(\lambda O_k D O_k{}^T\right)^{-1} W_k\right) + \sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}\log\left|\lambda O_k D O_k{}^T\right|$$

$$= \frac{1}{\lambda}\sum_{k=1}^{g} tr\left(W_k O_k D^{-1} O_k{}^T\right) + np\log(\lambda)$$

$$= \frac{1}{\lambda}\sum_{k=1}^{g} tr\left(O_k{}^T W_k O_k D^{-1}\right) + np\log(\lambda)$$

Let $W_k = L_k \Omega_k L_k{}^T$ be its eigenvalue decomposition, then

$$F_5 = \frac{1}{\lambda}\sum_{k=1}^{g} tr\left(O_k{}^T L_k \Omega_k L_k{}^T O_k D^{-1}\right) + np\log(\lambda)$$

Now F_5 can be written as $\frac{1}{\lambda}\sum_{k=1}^{g} tr\left(Q_k A Q_k{}^T B^{-1}\right) + np\log(\lambda)$, where $A = \Omega_k$, $B = D^{-1}$, and $Q_k = O_k{}^T L_k$ is an orthogonal matrix. F_5 can be minimized by minimizing each of $tr\left(Q_k A Q_k{}^T B^{-1}\right)$.

By Theorem 2, the minimum occurs when $Q_k = I$ or $O_k = L_k$. In this case $tr\left(Q_k A Q_k{}^T B^{-1}\right) = tr\left(AB^{-1}\right) = tr\left(\Omega_k D^{-1}\right)$ and so

$$F_5 = \frac{1}{\lambda}\sum_{k=1}^{g} tr\left(\Omega_k D^{-1}\right) + np\log(\lambda) = \frac{1}{\lambda} tr\left(\left(\sum_{k=1}^{g}\Omega_k\right)D^{-1}\right) + np\log(\lambda)$$

By Corollary 1, minimizing F_5 with respect to D and λ are

$$\hat{D} = \frac{\sum_{k=1}^{g}\Omega_k}{\left|\sum_{k=1}^{g}\Omega_k\right|^{1/p}}$$

and

$$\hat{\lambda} = \frac{\left|\sum_{k=1}^{g}\Omega_k\right|^{1/p}}{n}$$

3.2.2.6 Structure $\Sigma_k = \lambda_k O_k D O_k{}^T$

This is the structure where every covariance has same shape, but with different orientation and volume.

$F\left(\Sigma \mid x_1,...,x_n,\tau^*\right)$

$$= F_6\left(\lambda_1,\ldots,\lambda_g,O_1,\ldots,O_g,D \mid x_1,\ldots,x_n,\tau^*\right)$$

$$= \sum_{k=1}^{g} tr\left(\left(\lambda_k O_k D O_k^T\right)^{-1} W_k\right) + \sum_{k=1}^{g}\sum_{i=1}^{n} \tau_{ik} \log\left|\lambda_k O_k D O_k^T\right|$$

$$= \sum_{k=1}^{g} \frac{1}{\lambda_k} tr\left(W_k O_k D^{-1} O_k^T\right) + p\sum_{k=1}^{g}\sum_{i=1}^{n} \tau_{ik} \log\left(\lambda_k\right)$$

Since $W_k = L_k \Omega_k L_k^T$, the optimal λ_k, O_k, D are solutions of the equations to be solved iteratively

$$\widehat{\lambda}_k = \frac{tr\left(W_k O_k D^{-1} O_k^T\right)}{p\tau_k}$$

and

$$\widehat{O}_k = L_k$$

By Corollary 2, let $Q = \sum_{k=1}^{g} \frac{1}{\lambda_k} \Omega_k$, we get

$$\widehat{D} = \frac{\sum_{k=1}^{g} \frac{1}{\lambda_k} \Omega_k}{\left|\sum_{k=1}^{g} \frac{1}{\lambda_k} \Omega_k\right|^{1/p}}$$

3.2.2.7 Structure $\Sigma_k = \lambda_k O D_k O^T = O A_k O^T$, where $A_k = \lambda_k D_k$

Notice that $\left|A_k\right| = \left|\Sigma_k\right|$, this is the structure where every covariance has the same shape and volume, but with different orientation.

$$F\left(\Sigma \mid x_1,\ldots,x_n,\tau^*\right)$$

$$= F_7\left(O, A_1,\ldots,A_g \mid x_1,\ldots,x_n,\tau^*\right)$$

$$= \sum_{k=1}^{g} tr\left(O A_k^{-1} O^T W_k\right) + \sum_{k=1}^{g}\sum_{i=1}^{n} \tau_{ik} \log\left|A_k\right|$$

$$= \sum_{k=1}^{g} tr\left(A_k^{-1} O^T W_k O\right) + \sum_{k=1}^{g}\sum_{i=1}^{n} \tau_{ik} \log\left|A_k\right|$$

The minimization of F_7 has to be performed as follows.

1. For fix O, minimizing F_7 with respect to A_j is equivalent to minimizing

$$tr\left(A_k^{-1} O^T W_k O\right) + \sum_{i=1}^{n} \tau_{ik} \log\left|A_k\right|$$

By Corollary 2, let $a = \sum_{i=1}^{n} \tau_{ij} = \tau_j$, $M = A_j$, and $Q = O^T W_k O$. For fixed O, minimizing F_7 with respect to A_j, we get:

$$\widehat{A}_j = \frac{1}{\tau_j} diag\left(O^T W_j O\right), \quad j = 1,\ldots,g$$

2. For fixed $A_k = \lambda_k D_k = \lambda_k diag\left(\alpha_{1k},\ldots,\alpha_{pk}\right)$, for $k = 1,\ldots,g$ minimizing F_7 with respect to O is equivalent to minimizing

$$h(O) = \sum_{k=1}^{g} tr\left(W_k O A_k^{-1} O^T\right) = \sum_{k=1}^{g} tr\left(O^T W_k O A_k^{-1}\right)$$

Which is shown can be done by a variant of algorithm of Flury (1986) as follows:

Step 1: Start with initial solution matrix $O = \left(o_1,\ldots,o_p\right)$, where $o_1,\ldots,o_p$ are mutually orthonormal.

Step 2: For any indices $l, m \in \{1,\ldots,p\}$, where $l \neq m$, the pair $\left(o_l, o_m\right)$ is replaced with any pair $\left(\delta_l, \delta_m\right)$ where δ_l and δ_m are orthonormal vectors, each a linear combination of o_l and o_m, minimizing the above criterion $h(O)$. This can be obtained by the following procedure:

We have

$$= \sum_{k=1}^{g} tr\left(O^T W_k O A_k^{-1}\right)$$

$$= \sum_{k=1}^{g} tr\left(\left(o_1,\ldots,o_p\right)^T W_k \left(o_1,\ldots,o_p\right)\lambda_k^{-1} diag\left(\alpha_{1k}^{-1},\ldots,\alpha_{pk}^{-1}\right)\right)$$

$$= \sum_{k=1}^{g} tr\left(\left(o_1,\ldots,o_p\right)^T W_k \left(\frac{1}{\lambda_k \alpha_{1k}} o_1,\ldots,\frac{1}{\lambda_k \alpha_{pk}} o_p\right)\right)$$

$$= \sum_{k=1}^{g} \sum_{j=1}^{p} \left(\frac{o_j^T W_k o_j}{\lambda_k \alpha_{jk}}\right)$$

$$= \sum_{k=1}^{g} \left(\frac{o_l^T W_k o_l}{\lambda_k \alpha_{lk}}\right) + \sum_{k=1}^{g} \left(\frac{o_m^T W_k o_m}{\lambda_k \alpha_{mk}}\right) + \sum_{k=1}^{g} \sum_{j \neq l,m}^{p} \left(\frac{o_j^T W_k o_j}{\lambda_k \alpha_{jk}}\right)$$

$$= S\left(o_l, o_m\right) + \sum_{k=1}^{g} \sum_{j \neq l,m}^{p} \left(\frac{o_j^T W_k o_j}{\lambda_k \alpha_{jk}}\right),$$

where $S\left(o_l, o_m\right) = \sum_{k=1}^{g} \left(\frac{o_l^T W_k o_l}{\lambda_k \alpha_{lk}}\right) + \sum_{k=1}^{g} \left(\frac{o_l^T W_k o_l}{\lambda_k \alpha_{mk}}\right)$

Now we introduce new orthonormal vectors $\left(\delta_l, \delta_m\right)$ to replace as follows

$$\delta_l = \left(o_l, o_m\right) q_1,$$

and

$$\delta_m = \left(o_l, o_m\right) q_2,$$

where q_1 and q_2 are vectors to be determined

Note that:

$$1 = \delta_l{}^T \delta_l = q_1{}^T (o_l, o_m)^T (o_l, o_m) q_1 = q_1{}^T \begin{bmatrix} o_l^T o_l & o_l^T o_m \\ o_m^T o_l & o_m^T o_m \end{bmatrix} q_1 = q_1{}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} q_1 = q_1{}^T q_1$$

and

$$0 = \delta_l{}^T \delta_m = q_1{}^T (o_l, o_m)^T (o_l, o_m) q_2 = q_1{}^T \begin{bmatrix} o_l^T o_l & o_l^T o_m \\ o_m^T o_l & o_m^T o_m \end{bmatrix} q_2 = q_1{}^T \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} q_2 = q_1{}^T q_2$$

and so q_1, q_2 are two orthonormal vectors as well.

We choose q_1, q_2 to minimize $S(\delta_l, \delta_m)$

$$S(\delta_l, \delta_m) = \sum_{k=1}^{g} \left(\frac{q_1{}^T (o_l, o_m)^T W_k (o_l, o_m) q_1}{\lambda_k \alpha_{lk}} \right) + \sum_{k=1}^{g} \left(\frac{q_2{}^T (o_l, o_m)^T W_k (o_l, o_m) q_2}{\lambda_k \alpha_{mk}} \right)$$

Let $Y_k = (o_l, o_m)^T W_k (o_l, o_m)$, then we can write

$$S(\delta_l, \delta_m) = \sum_{k=1}^{g} \left(\frac{q_1{}^T Y_k q_1}{\lambda_k \alpha_{lk}} \right) + \sum_{k=1}^{g} \left(\frac{q_2{}^T Y_k q_2}{\lambda_k \alpha_{mk}} \right)$$

Denote $Q = (q_1, q_2)$, we get $q_1{}^T Y_k q_1 + q_2{}^T Y_k q_2 = tr(Q^T Y_k Q) = tr(Y_k)$

And the problem reduces to the optimization of

$$S(\delta_l, \delta_m) = \sum_{k=1}^{g} \left(\frac{q_1{}^T Y_k q_1}{\lambda_k \alpha_{lk}} \right) + \sum_{k=1}^{g} \left(\frac{tr(Y_k) - q_1{}^T Y_k q_1}{\lambda_k \alpha_{mk}} \right)$$

which is equivalent to the minimization of

$$q_1^T \left\{ \sum_{k=1}^{g} \left(\frac{1}{\lambda_k \alpha_{lk}} - \frac{1}{\lambda_k \alpha_{mk}} \right) Y_k \right\} q_1,$$

So q_1 is the second eigenvector of the matrix $\sum_{k=1}^{g} \left(\frac{1}{\lambda_k \alpha_{lk}} - \frac{1}{\lambda_k \alpha_{mk}} \right) Y_k$,

Repeat step 2 until produces no decrease of the criterion.

3.2.2.8 Structure $\Sigma_k = \lambda_k O_k D_k O_k^T$

This is the structure where every covariance has different shape, different volume, and different orientation.

$$F\left(\Sigma \mid x_1,...,x_n,\tau^*\right)$$

$$= F_8\left(\Sigma_1,...,\Sigma_g \mid x_1,...,x_n,\tau^*\right)$$

$$= \sum_{k=1}^{g} tr\left(W_k \Sigma_k^{-1}\right) + \sum_{k=1}^{g}\sum_{i=1}^{n} \tau_{ik} \log\left|\Sigma_k\right|$$

We already know, $\hat{\Sigma}_k = \frac{W_k}{\sum_{i=1}^{n} \tau_{ik}} = \frac{W_k}{\tau_k}$, and so we can get the estimates λ_k, O_k and D_k by eigenvalue decomposition.

Table 2: Parameter estimation for geometric covariate matrix decomposition

Structure	Σ_k	$F(\Sigma \mid x_1,...,x_n,\tau^*)$	Solution
1	λI	$\lambda^{-1}tr(W)+np\log(\lambda)$	$\hat{\lambda}=\frac{tr(W)}{np}$
2	$\lambda_k I$	$\sum_{k=1}^{g}\lambda_k^{-1}tr(W_k)+p\sum_{k=1}^{g}\log(\lambda_k)\tau_k$	$\hat{\lambda}_k=\frac{tr(W_k)}{p\tau_k}$
3	Σ	$tr(W\Sigma^{-1})+n\log\lvert\Sigma\rvert$	$\hat{\Sigma}=\frac{W}{n}$
4	$\lambda_k\sum_0$	$\sum_{k=1}^{g}\lambda_k^{-1}tr(W_kC^{-1})+p\sum_{k=1}^{g}\log(\lambda_k)\tau_k$	Iterative procedure
5	$\lambda O_k D O_k^T$	$\frac{1}{\lambda}tr(\Omega_k D^{-1})+np\log(\lambda)$	$\hat{\lambda}=\frac{\left\lvert\sum_{k=1}^{g}\Omega_k\right\rvert^{1/p}}{n}$ $\hat{O}_k=L_k$ $\hat{D}=\frac{\sum_{k=1}^{g}\Omega_k}{\left\lvert\sum_{k=1}^{g}\Omega_k\right\rvert^{1/p}}$
6	$\lambda_k O_k D O_k^T$	$\sum_{k=1}^{g}\frac{1}{\lambda_k}tr(W_kO_kD^{-1}O_k^T)+p\sum_{k=1}^{g}\log(\lambda_k)\tau_k$	$\hat{\lambda}_k=\frac{tr(W_kO_kD^{-1}O_k^T)}{p\tau_k}$ $\hat{O}_k=L_k$ $\hat{D}=\frac{\sum_{k=1}^{g}\frac{1}{\lambda_k}\Omega_k}{\left\lvert\sum_{k=1}^{g}\frac{1}{\lambda_k}\Omega_k\right\rvert^{1/p}}$
7	$\lambda_k O D_k O^T$	$\sum_{k=1}^{g}tr(W_kOA_k^{-1}O^T)+\sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}\log\lvert A_k\rvert$	Iterative procedure
8	Σ_k	$\sum_{k=1}^{g}tr(W_k\Sigma_k^{-1})+\sum_{k=1}^{g}\sum_{i=1}^{n}\tau_{ik}\log\lvert\Sigma_k\rvert$	$\hat{\Sigma}_k=\frac{W_k}{\tau_k}$

3.2.3 EM procedure in finite MVN mixture

The EM algorithm for a mixture of multivariate normal can be expressed as the following procedure:

Initialize $\Theta^{(0)}$. Take $\underline{\pi}^{(0)} = (\pi_1^{(0)}, \pi_2^{(0)}, \ldots, \pi_g^{(0)}) = \left(\frac{1}{g}, \frac{1}{g}, \ldots, \frac{1}{g}\right)$. The kth mean vector $\underline{u_k}^{(0)} = \underline{\bar{x}}$, the overall sample mean for all group. The kth covariance matrix $\sum_k^{(0)} = S$, the overall sample covariance matrix for all group.

E-Step: A conditional expectation of the group membership for each observation can be evaluated by calculating

$$\tau_{ik}^{(t)} = E\left(z_{ik}^{(t)} \mid x_i ; \Theta^{(t)}\right) \overset{Bayes}{\overbrace{=}} \frac{\pi_k^{(t-1)} \phi\left(\underline{x_i} \mid \underline{u_k}^{(t-1)}, \sum_k^{(t-1)}\right)}{\sum_{k=1}^{K} \pi_k^{(t-1)} \phi\left(\underline{x_i} \mid \underline{u_k}^{(t-1)}, \sum_k^{(t-1)}\right)}.$$

M-Step: Compute sufficient statistics by

$$T_{k1}^{(t)} = \sum_{i=1}^{n} \tau_{ik}^{(t)}, \quad T_{k2}^{(t)} = \sum_{i=1}^{n} \tau_{ik}^{(t)} x_i \quad \text{and} \quad T_{i3}^{(t)} = \sum_{i=1}^{n} \tau_{ik}^{(t)} x_i x_i^{T}.$$

Get the parameter estimates $\widehat{\Theta}^{(t)}$ by

$$\hat{\pi}_k^{(t)} = T_{k1}^{(t)} / n \quad , \quad \underline{\hat{u}_k}^{(t)} = T_{k2}^{(t)} / T_{k1}^{(t)} \quad \text{and} \quad \hat{\Sigma}_k^{(t)} = \left\{ T_{k3}^{(t)} - T_{k1}^{(t)^{-1}} T_{k2}^{(t)} T_{k2}^{(t)^{T}} \right\} / T_{k1}^{(t)}$$

Loop back to the E-step until the following convergence criteria are met at once

$$\begin{cases} \hat{\pi}_k^{(t)} - \hat{\pi}_k^{(t-1)} <= \textit{threshold}, \text{ for k= 1,…,g} \\ \underline{\hat{u}_k}^{(t)} - \underline{\hat{u}_k}^{(t-1)} <= \textit{threshold}, \text{ for k=1,…,g} \\ \hat{\Sigma}_{ij}^{(t)} - \hat{\Sigma}_{ij}^{(t-1)} <= \textit{threshold}, \text{ for k=1,…,g and any ij conbination} \end{cases}$$

Notice that the M-Step can use the result of section 3.2.2 if a parsimonious decomposition is considered.

Here, the first phase is to choose a good starting point. A poor starting point might make the convergence process very slow or even make the sequence of estimates diverge if it is chosen too close to the boundary. A good starting point would be one, which led to faster convergence and avoided local maxim. There is also no guarantee that the maximum achieved is global.

When there is no a priori good choice for a starting value, the start is sometimes chosen at random or perhaps from the output of a clustering procedure. Suppose for example k-groups are then produced since it can be interpreted as a parsimonious model, a simple independent Gaussian distribution with equal volume spherical covariance structure. In the clustering context, the EM algorithm for mixture models is usually initialized with a hierarchical clustering step (Dasgupta and Raftery, 1998; Fraley and Raftery, 1998).

The threshold stopping criterion needs to be set small enough to make sure the maximum value is obtained. In choosing a threshold, there is a trade-off between convergence time and accuracy, since EM will converge very quickly and will converge faster in the beginning. A better accuracy will definitely need extra step to run the program.

4. Gibbs Sampler

4.1 Gibbs in general

Estimation in a Bayesian framework is also feasible using posterior simulation via Markov Chain Monte Carlo (MCMC) method. Bayes estimators for mixture models are well defined as long as the conjugate prior distributions are proper.

For each model $\Theta = (\Theta_1, \Theta_2, ..., \Theta_d)$, we define a neighbourhood $nbd(\Theta)$ consisting of Θ itself and the models which differ from Θ by just one parameter. A transition probability T is defined by setting $T(\Theta \to \Theta') = 0$ for all $\Theta' \notin nbd(\Theta)$, and $T(\Theta \to \Theta')$ constant for all $\Theta' \in nbd(\Theta)$.

The Gibbs Sampler, formally introduced by Geman and Geman in 1984, is an algorithm for extracting the marginal distribution from the conditional distribution. We need to draw a sample of the parameters from the full mixture distribution function, which we do not know how to do. However we do know how to draw a sample from the conditional distribution function of each parameter given the others.

Start with the current state $\Theta = (\Theta_1, \Theta_2, ..., \Theta_d)$ in d-dimensional space, and consider a new state N in the chain

$$N = (\Theta_1, \Theta_2, ..., \Theta_{q-1}, N_q, \Theta_{q+1}, ..., \Theta_d)$$

where the only difference between state Θ and state N is the value of the qth parameter in the vector with all others the same. The conditional probability of N_q given $\Theta_{-q} = (\Theta_1, \Theta_2, ..., \Theta_{q-1}, \Theta_{q+1}, ..., \Theta_d)$ will be written as

$$\pi\left(N_q \mid \Theta_{-q}\right) = \pi\left(N_q \mid \Theta_1, \Theta_2, \ldots, \Theta_{q-1}, \Theta_{q+1}, \ldots, \Theta_d\right)$$

The transition probabilities are

$$T(\Theta \to N) = \pi\left(N_q \mid \Theta_{-q}\right)$$

and

$$T(N \to \Theta) = \pi\left(\Theta_q \mid N_{-q}\right) = \pi\left(\Theta_q \mid \Theta_{-q}\right), \text{ since } N_{-q} = \Theta_{-q}$$

Consider drawing the sample $\Theta = (\Theta_1, \Theta_2, \ldots, \Theta_d)$ from joint probability function $\pi(\Theta)$. The detail process of getting the parameter $\Theta = (\Theta_1, \Theta_2, \ldots, \Theta_d)$ using Gibbs sampler is as follows:

Start with an initial value $\Theta^{(0)} = \left(\Theta_1^{(0)}, \Theta_2^{(0)}, \ldots, \Theta_d^{(0)}\right)$, for each iteration we execute the following d step as follows

Step 1: Draw sample $\Theta_1^{(k+1)} \sim f\left(\Theta_1 \mid X; \Theta_2^{(k)}, \ldots, \Theta_d^{(k)}\right)$

Step 2: Draw sample $\Theta_2^{(k+1)} \sim f\left(\Theta_2 \mid X; \Theta_1^{(k+1)}, \Theta_3^{(k)}, \ldots, \Theta_d^{(k)}\right)$

…

Step d: Draw sample $\Theta_d^{(k+1)} \sim f\left(\Theta_d \mid X; \Theta_1^{(k+1)}, \Theta_2^{(k+1)}, \ldots, \Theta_{d-1}^{(k+1)}\right)$

Continue the above procedure, we get the Gibbs sampler after a burn-in process. Madigan and York (1992) reported that this process is highly mobile and that runs of 10,000 or less are typically adequate. The vector sequence $\Theta = (\Theta_1, \Theta_2, \ldots, \Theta_d)$ thus generated is known to be a realization of a homogeneous Markov Chain. The above procedure was proven to converge in distribution to the true posterior distribution of $\Theta = (\Theta_1, \Theta_2, \ldots, \Theta_d)$ by Diebolt and Robert (1994). Raftery (1992) gave a method for determining the number of iterations to be dropped in the burn-in process and the minimum number of iterations needed to be run beyond burn-in. All steps beyond burn-in provide data which can be used to produce an estimate (e.g. histogram) of the joint distribution.

Here, all parameters are treated as random variables with joint probability function $f(\Theta)$. Instead of estimating these parameters by EM, we can approximate the parameter distribution by building the histogram of a sample. This joint distribution sample approximation can be approached by sampling from the conditional distributions. It turns out that after discarding the first initial sample, the following sample is an approximation of sample from the joint distribution function.

4.2 Gibbs in MVN

4.2.1 Posterior distribution

If the distribution is a multivariate normal mixture of g components, the parameter vector is

$$\{\pi_1,\ldots,\pi_{g-1},\underline{u_1},\ldots,\underline{u_g},\Sigma_1,\ldots,\Sigma_g\}$$

Simulate the unknown joint distribution $f(\pi_1,\ldots,\pi_{g-1},\underline{u_1},\ldots,\underline{u_g},\Sigma_1,\ldots,\Sigma_g)$ by simulating the conditional distribution function as follows:

$$\underline{\pi} \sim f(\underline{\pi} \mid X;\underline{u_1},\ldots,\underline{u_g},\Sigma_1,\ldots,\Sigma_g)$$

$$u_j \sim f(u_i \mid X;\pi_1,\ldots,\pi_{g-1},\underline{u_1},\ldots,u_{j-1},u_{j+1}\ldotp\ldotp,\underline{u_g},\Sigma_1,\ldots,\Sigma_g)$$

$$\Sigma_j \sim f(\Sigma_i \mid X;\pi_1,\ldots,\pi_{g-1},\underline{u_1},\ldotp\ldotp,\underline{u_g},\Sigma_1,\ldots,\Sigma_{j-1},\Sigma_{j+1}\ldots,\Sigma_g)$$

With prior distribution: $\underline{\pi} \mid \underline{\alpha} \sim Dirichlet(\alpha_1, \alpha_1, .., \alpha_g)$, the conditional proportion is defined as $f(\underline{\pi} \mid \underline{z_i}) = Dirichlet(\alpha_1 + n_1, \alpha_2 + n_2, ..., \alpha_g + n_g)$, where z_i is the indicator vector defined as before.

Let ξ_k be the kth mean vector, $\tau_k = \sum_{i=1}^{n} \tau_{ik}$, and $\overline{\underline{\xi_k}} = (n_k \underline{u_k} + \tau_k \underline{\xi_k}) / (n_k + \tau_k)$. The conditional distribution for group mean and covariance matrix are given by Diebolt and Robert (1994), see also Bensmail et al (1996), which is shown in table 3 and table 4.

4.2.2 Gibbs sampling in finite MVN mixture

Initialize $\Theta^{(0)}$ using the same procedure of EM.

Set $\alpha_k = 1$, where $k = 1, ..., g$. And $\underline{\pi}^{(0)} = (\pi_1^{(0)}, \pi_2^{(0)}, ..., \pi_g^{(0)}) = \left(\frac{1}{g}, \frac{1}{g}, ..., \frac{1}{g}\right)$.

Bayesian Step: Calculate

$$\tau_{ik}^{(t)} \overset{\overbrace{}^{Bayes}}{=} \frac{\pi_k^{(t-1)} \phi\left(\underline{x_i} \mid \underline{u_k}^{(t-1)}, \Sigma_k^{(t-1)}\right)}{\sum_{k=1}^{K} \pi_k^{(t-1)} \phi\left(\underline{x_i} \mid \underline{u_k}^{(t-1)}, \Sigma_k^{(t-1)}\right)}, \text{ for i=1, …, n,}$$

where τ_{ik} is the probability the observation i belong to group k, given current parameters

Simulation Step:

[1] Simulate each $\underline{z_i}$ according to uniform distribution

$$\Pr\left(\underline{z_i}^{(t)} \mid \tau_{ik}^{(t)}\right) = Unif\left(\tau_{i1}^{(t)}, \tau_{i1}^{(t)}, .., \tau_{ig}^{(t)}, \right), \text{ for i=1,…n,}$$

[2] Simulate $\underline{\pi}^{(t)} = \left(\pi_1^{(t)}, \pi_2^{(t)}, ..., \pi_g^{(t)}\right)$ according to

$$\Pr\left(\underline{\pi}^{(t)} \mid \underline{z_i}^{(t)}\right) = Dirichlet\left(\alpha_1 + n_1, \alpha_2 + n_2, \ldots, \alpha_g + n_g\right)$$

[3] Simulate $\underline{\theta}^{(t)} = \left(u_1^{(t)},..,u_g^{(t)}; \sum_1^{(t)},..,\sum_g^{(t)}\right)$ according to $\Pr\left(\underline{\theta}^{(t)} \mid \underline{z_i}^{(t)}\right)$

a. Simulate $\underline{u_k}^{(t)} \mid \sum_k^{(t)}, \underline{z}^{(t)} \sim$ from the corresponding conditional distribution from table 3.

b. Simulate $\sum_k^{(t)} \mid \underline{z}^{(t)} \sim$ from the corresponding conditional distribution from table 4.

Loop back to Bayesian step until the following convergence criteria are met together

$$\begin{cases} & \text{for all k} \\ u_k^{(t)} - u_k^{(t-1)} <= threshold & \text{for all k} \\ \Sigma_{ij}^{(t)} - \Sigma_{ij}^{(t-1)} <= threshold & \text{for all ij combination} \end{cases}$$

Note: the conditional distribution only updates one parameter each time conditional on all other parameters are fixed, and need to be simulated according to the sequence above.

Based on different model assumptions, we have 5 common methods to do the simulation.

Table 3: Probability Distributions of Mean Vector

Structure	Σ_k	Prior Distribution	Posterior Distribution Conditional on $\underline{Z}$
1	λI	$\underline{u_k} \mid \lambda \sim N_p\left(\xi_k, \frac{\lambda}{\tau_k} I_p\right)$	$\underline{u_k} \mid \lambda, \underline{Z} \sim N_p\left(\overline{\xi}_k, \frac{\lambda}{n_k + \tau_k} I_p\right)$
2	$\lambda_k I$	$\underline{u_k} \mid \lambda_k \sim N_p\left(\xi_k, \frac{\lambda_k}{\tau_k} I_p\right)$	$\underline{u_k} \mid \lambda_k, \underline{Z} \sim N_p\left(\overline{\xi}_k, \frac{\lambda_k}{n_k + \tau_k} I_p\right)$
3	Σ	$\underline{u_k} \mid \Sigma \sim N_p\left(\xi_k, \frac{1}{\tau_k} \Sigma\right)$	$\underline{u_k} \mid \lambda_k, \underline{Z} \sim N_p\left(\overline{\xi}_k, \frac{\lambda_k}{n_k + \tau_k} \Sigma\right)$

4	$\lambda_k \Sigma$	$\underline{u_k} \mid \Sigma_0 \sim N_p\left(\xi_k, \frac{\lambda_k}{\tau_k}\Sigma_0\right)$	$\underline{u_k} \mid \lambda_k, \Sigma_0, \underline{Z} \sim N_p\left(\overline{\xi}_k, \frac{\lambda_k}{n_k + \tau_k}\Sigma_0\right)$
5-8	Σ_k	$\underline{u_k} \mid \sum_k \sim N_p\left(\xi_k, \frac{1}{\tau_k}\Sigma_k\right)$	$\underline{u_k} \mid \sum_k, \underline{Z} \sim N_p\left(\overline{\xi_k}, \frac{\Sigma_k}{\tau_k + n_k}\right)$

Table 4: Probability Distributions of Covariance Matrix

Structure	Σ_k	Prior Distribution	Posterior Distribution Conditional on $\underline{Z}$
1	λI	$\lambda_k \sim Ig\left(\frac{1}{2}m_0, \frac{1}{2}s_0^2\right)$	$\lambda \mid \underline{Z} \sim Ig\left(\frac{m_0 + n}{2}, \frac{1}{2}\left\{s_0^2 + \sum_k tr(W_k) + \sum_k \frac{n_k\tau_k}{n_k + \tau_k}(u_k - \xi_k)'(u_k - \xi_k)\right\}\right)$
2	$\lambda_k I$	$\lambda_k \sim Ig\left(\frac{1}{2}m_k, \frac{1}{2}s_k^2\right)$	$\lambda_k \mid \underline{Z} \sim Ig\left(\frac{m_0 + n_k p}{2}, \frac{1}{2}\left\{s_k^2 + tr(W_k) + \frac{n_k\tau_k}{n_k + \tau_k}(u_k - \xi_k)'(u_k - \xi_k)\right\}\right)$
3	Σ	$\Sigma \sim W_p^{-1}(m_0, \Psi_0)$	$\Sigma \mid \underline{Z} \sim W_p^{-1}\left(m_0 + n, \Psi_0 + \sum_k \left\{W_k + \frac{n_k\tau_k}{n_k + \tau_k}(u_k - \xi_k)'(u_k - \xi_k)\right\}\right)$
4	$\lambda_k \sum$	$\lambda_k \sim Ig\left(\frac{1}{2}r_k, \frac{1}{2}\rho_k\right)$	$\lambda_k \mid \Sigma_0, \underline{Z} \sim Ig\left(\frac{r_k + n_k p}{2}, \frac{1}{2}\left\{\rho_k + tr(W_k\Sigma_0^{-1}) + \frac{n_k\tau_k}{n_k + \tau_k}(u_k - \xi_k)'\Sigma_0^{-1}(u_k - \xi_k)\right\}\right)$
		$\Sigma_0 \sim W_p^{-1}(m_0, \Psi_0)$	$\Sigma_0 \mid \underline{\lambda}, \underline{Z} \sim W_p^{-1}\left(m_0 + n, \Psi_0 + \sum_k \left\{\frac{1}{\lambda_k}W_k + \frac{n_k\tau_k}{\lambda_k(n_k + \tau_k)}(u_k - \xi_k)'(u_k - \xi_k)\right\}\right)$
5-8	Σ_k	$\sum_k \sim W_p^{-1}(m_k, \Psi_k)$	$\sum_k \mid \underline{Z} \sim W_p^{-1}\left(m_k + n_k, \Psi_k + W_k + \frac{n_k\tau_k}{(n_k + \tau_k)}(u_k - \xi_k)'(u_k - \xi_k)\right)$

where $Ig\left(\frac{1}{2}r, \frac{1}{2}\rho\right)$ is Inverse Gamma distribution function, and $W_p^{-1}(\sum, n)$ is Inverse Wishart distribution function.

5. Model-based Clustering

Suppose the number of cluster is known, the parameters are fit, and we can cluster by EM algorithm. In the end of the iterative procedure, we have the convergence value for all parameters and the fitted posterior probabilities $\tau_{i1}, \tau_{i2}, ..., \tau_{ig}$ for component membership probability of each observation. Then each observation is assigned to the group with the maximum conditional probability, which can be accessed by the component label vector $\underline{z_i} = (z_{i1}, z_{i2}, ..., z_{ig})$, as defined below

$$z_{ik}^{(t)} = 1, \text{ if } \tau_{ik}^{(t)} = \max\{\tau_{i1}^{(t)}, \tau_{i1}^{(t)}, .., \tau_{ig}^{(t)}\}$$

$$= 0, \text{ otherwise}$$

All that needs is to choose the parameter structure as table 1. This all works if g is known but what if g is unknown. We will introduce how to select the model as follows.

5.1 Model selection

5.1.1 General procedure

But the problem is that EM algorithm works only when the number of cluster is specified, which is the basic assumption. That is to measure the probability $f(X \mid \Theta_k)$ given M_k (model k), where k is the index of model. Since both the number of cluster and the model need to be specified to run EM for clustering. We need to find a way to select the number of cluster and model together. That is to measure the integrated likelihood $f(\Theta_k \mid X)$ instead of just $f(X \mid \Theta_k)$ for a specific model. Here Banfield and Raftery proposed to use 8 covariance structures for 8 kinds of models, and this may be extended to more general case.

In Banfield and Raftery's approach, each combination of different specification of the covariance matrix and different number of clusters corresponds to a separate probability model. The probabilistic framework of model-based clustering allows the issues of choosing the best clustering model and the correct number of clusters to be reduced simultaneously to a model selection problem. This is important because there is a trade-off between probability model (and the corresponding clustering method) and number of clusters. It is easy to see that a complex model only needs a small number of clusters, but a simple model may need a larger number of clusters to fit the data adequately.

Suppose that K models, $M_1, M_2, ..., M_K$, are being considered, each a different model with a different number of clusters and parameters. Take a simple example of comparing two models M_i and M_j, here i and j are model indexes. In order to access the model M_i and M_j, we measure the posterior probability of different models given data X. Then, by Bayes' theorem, the posterior probability of M_i is

$$f(M_i \mid X) = f(X \mid M_i) f(M_i) / \sum_{k=1}^{K} f(X \mid M_k) f(M_k),$$

and the posterior odds for comparing model M_i to model M_j is

$$\frac{f(M_i \mid X)}{f(M_j \mid X)} = \frac{f(X \mid M_i) f(M_i) / \sum_{k=1}^{K} f(X \mid M_k) f(M_k)}{f(X \mid M_j) f(M_j) / \sum_{k=1}^{K} f(X \mid M_k) f(M_k)}$$

$$= \frac{f(X \mid M_i) f(M_i)}{f(X \mid M_j) f(M_j)}$$

When all models are assumed to have equal prior probabilities, the odds become $B_{ij} = f(X \mid M_i)/f(X \mid M_j)$, which is the Bayes factor defined by Kass and Raftery (1995) as the ratio of the integrated likelihoods of the two models $B_{ij} = f(X \mid M_i)/f(X \mid M_j)$. In other words, the Bayes factor B_{ij} represents the posterior odds that the data are distributed according to model M_i against model M_j, assuming that neither model is favoured a priori (ie. $f(M_i) = f(M_j)$). If B_{ij} > 1, model M_i is favoured over M_j. The method can be easily generalized to more than two models.

To determine the Bayes factor, we require the integrated likelihood of model M_k, given as

$$f(X \mid M_k) = \int f(X \mid \Theta_k, M_k) f(\Theta_k \mid M_k) d\Theta_k$$

where $k = i, j$ and $f(\Theta_k \mid M_k)$ is the prior distribution of Θ_k. This integrated likelihood represents the probability that data X is observed given that the underlying model is M_k. Two approaches are considered in the evaluation of the integrated likelihood.

5.1.2 Approach 1 - Bayesian Information Criterion

Here, we use an approximation called the Bayesian Information Criterion (BIC; Schwarz, 1978)

$$2\log f(X \mid M_k) \approx 2\log f(X \mid \hat{\Theta}_k, M_k) - v_k \log(n) = BIC$$

where v_k is the number of parameters to be estimated in Model M_k, and $\hat{\Theta}_k$ is the maximum likelihood estimate of parameter vector Θ_k. Intuitively, the first term, which is the maximized mixture likelihood for the model, rewards a model that fits the data well, and the second term discourages over fitting by penalizing modes with more free parameters.

Hence, we get $\log(B_{ij}) = \log f(X \mid M_i) - \log f(X \mid M_j) = \frac{1}{2}(BIC_i - BIC_j)$

If $BIC_i > BIC_j$ then $\log(B_{ij}) > 0$, ie. $B_{ij} > 1$. Model i is better than model j.

A large BIC score indicates strong evidence for the corresponding model. The BIC score can be used to compare models with different covariance matrix parameterizations and different numbers of clusters. Usually, BIC score differences greater than 10 are considered as strong evidence favoring one model over another (Kass and Raftery, 1995).

5.1.3 Approach 2 - Laplace Approximation

$f(\Theta_k \mid M_k)$ can be calculated by the Laplace approximation (Tierney and Kadane 1986)

$$\int e^{g(\underline{u})} du \approx (2\pi)^{d/2} |A|^{\frac{1}{2}} \exp\{g(\underline{u}^*)\},$$

where $\underline{u}^*$ is the value of $\underline{u}$ at which g attains its maximum, and A is minus the inverse Hessian of g evaluated at $\underline{u}^*$.

Let $\underline{u} = \Theta_k$, $A = \psi$, $g(\underline{u}) = g(\Theta_k) = \log[f(X \mid \Theta_k, M_k) f(\Theta_k \mid M_k)]$ or

$$e^{g(\underline{u})} = e^{g(\Theta_k)} = f(X \mid \Theta_k, M_k) f(\Theta_k \mid M_k)$$

Apply the Laplace approximation to the above equation, yields

$$f(X \mid M_k) = \int f(X \mid \Theta_k, M_k) f(\Theta_k \mid M_k) d\Theta_k$$

$$\approx (2\pi)^{d/2} |\psi|^{\frac{1}{2}} f(X \mid \widehat{\Theta}_k, M_k) f(\widehat{\Theta}_k \mid M_k)$$

where d is the dimension of Θ_k, $\hat{\Theta}_k$ is the posterior mode of Θ_k, and ψ is minus the inverse Hessian of $g(\Theta_k) = \log\{f(X \mid \Theta_k)f(\Theta_k)\}$, evaluated at $\Theta_k = \hat{\Theta}_k$.

In many practical situations, an analytical solution is not available. Raftery (1996a) has suggested to use the Gibbs sampling, a special case of Metropolis-Hastings algorithm, to find the estimates for $\hat{\Theta}_k$ and $|\hat{\psi}|$ by using posterior simulation to estimate the quantities it needs. The whole procedure is called Laplace-Metropolis estimator. The Laplace method requires the posterior mode, $\hat{\Theta}_k$ and $|\hat{\psi}|$. To estimate $\hat{\Theta}_k$ from posterior simulation output, and probably the most accurate, is to compute $g(\Theta_k)^{(t)}$ for each t=1,…,T. and take the largest value, or just simply use the posterior mean or median. The matrix $\hat{\psi}$ is asymptotically equal to the posterior covariance matrix, as sample size tends to infinity, and so it would seem natural to approximate ψ by the estimated posterior covariance matrix from the posterior simulation output. To avoid the MCMC trajectories, Banfield and Raftery (1997) use the weighted covariance matrix estimated with weights based on the minimum volume ellipsoid estimate of Rousseeuw and van Zomeren (1990).

5.2 Clustering procedure

5.2.1 Clustering by EM

A comprehensive clustering strategy based on EM and Bayes factor is proposed by Banfield and Raftery (1993) as follows:

1. Set a maximum for the number of clusters, which is usually less that 10. A set of mixture models, say, a subset of 8 covariance structures is considered.

2. Perform hierarchical agglomeration to approximately maximize the classification likelihood for each model, and obtain the corresponding classifications for up to M groups.
3. Apply EM algorithm for each model and each number of clusters 2,…,M, starting with the classification result from hierarchical agglomeration.
4. Compute BIC for the one-cluster case for each model and for the mixture model with the optimal parameters from EM for 2, …, M clusters.
5. Choose the model corresponding to the largest BIC.

Although BIC works fairly well in practice, it is quite crude. More accurate approximation to Bayes factors can be obtained from Gibbs sampler output using the Laplace-Metropolis estimator (Raftery 1996). This is shown to give accurate results by Lewis and Raftery (1997).

5.2.2 Clustering by Gibbs sampling

Apply the same procedure as section 5.2.1, the Gibbs sampling output can used to choose the parameter and the number of cluster together. The algorithm is then as follows:

1. Set a maximum for the number of clusters, which is usually less that 10. A set of mixture models, say, a subset of 8 covariance structures is considered.
2. Perform hierarchical agglomeration to approximately maximize the classification likelihood for each model, and obtain the corresponding classifications for up to M groups.
3. Apply Gibbs algorithm for each model and each number of clusters 2,…,M, starting with the classification result from hierarchical agglomeration.
4. Compute integrated likelihood from Gibbs sampling output in last step. For the one-cluster case for each model and for the mixture model with the optimal

parameters from Gibbs sampling for 2, …, M clusters.

5. Choose the model corresponding to the largest integrated likelihood.

5.3 Clustering software

Some useful packages are available in statistical software R for free to download as follows:

5.3.1 MCLUST - Model-based cluster analysis

Fraley and Raftery implemented model-based clustering function called EMCLUST using EM-BIC approach. A list of useful function and corresponding syntax are as follows:

Hierarchical Clustering

Syntax: *hc* (modelName='EII', data, …)

By setting Model Name = "EII", the covariance structure is specified as spherical and equal volume, correspond to the first model of the table 1.

The initial parameter can be obtained by model-based hierarchical clustering by the function *hc*. By specifying the simplest model in hierarchical clustering. A class label can be used in the following m-step.

Parameterized gaussian mixture models

estep – individual E-step of EM algorithm

mstep – individual M-step of EM algorithm em - EM iterative (starting with e_step and then m-step and etc.)

me – EM iterative (starting with m-step and then e-stop and etc.)

Since EM algorithm can be start with either E-step (followed by M-step) or M-step (followed by E-step), the second step will be *mstep* (the function of individual M-step of EM algorithm followed by the result of *hc* function). Then run *mstep* function alternatively until converge. The above iterative procedure can also be substituted by a simple *me* function (EM algorithm starting with m-step).

Plotting functions

CoordProj(data, dimens, type, …)

clPairs (data, classification, symbols, label)

coordProj function coordinates projections of data in more than two dimensions modeled by an MVN mixture

clPairs function can creates a scatter plot for each pair of variables in given data, observations in different classes are represented by different symbols.

BIC for model-based clustering

Emclust (data, G, emModelName, hcPairs, subset, eps, tol, itmax, equalPro, warnSingular, …)

The whole procedure can also be run automatically by just use the *emclust* function. It can execute the above procedure in an integrated way, compare difference model and give users the best model and parameters.

5.3.2 MCMCpack - Markov chain Monte Carlo (MCMC) package

Some useful simulation functions:

rdirichlet (n, alpha): Generate the dirichlet distribution with n random vector with parameter alpha

rinvGamma(n, shape, scale=1): Generate the inverse gamma distribution with n draw and the scale shape parameter

riwish(v, S): The Inverse Wishart Distribution with v is the degree of freedom and S is pxp scale matrix

5.3.3 McGibbsit - Run-length diagnostic for Gibbs sampler

read.mcmc (nc, sourcepattern, col.name, start=1, end=nrow(tmp), thin=1)

mcgibbsit (data, q=0.025, r=0.0125, s=0.95, converge.eps=0.001, correct.cor=TRUE)

read.mcmc function can be used to read in data from a set of MCMC runs

mcgibbsit function can be used to diagnostic Functions for deciding the number of burn-in step and the stopping step are as follows:

6. References

[1] Bensmail, H., Celeux, G., Raftery, A. E. and Robert, C. P. (1997) Inference in model-based cluster analysis, *Statistics and Computing*, 7, 1-10

[2] Banfield, J. D. and Raftery, A. E. (1993) Model-based Gaussian and Non Gaussian Clustering. *Biometrics*, 49, 803-821

[3] Celeux, G. and G. Govaert (1995) Gaussian parsimonious clustering models. *Pattern Recognition*, 28, 781-793.

[4] C HIH -C HIEN Y ANG et al. (1999) Finite Mixture Multivariate Generalized Linear Models Using Gibbs Sampling and EM Algorithms, *Proc. Natl. Sci. Counc*. ROC(A) Vol.23, 6, 695-702

[5] Dempster, A. P., Laird, N. M. and Rubin, D. B. (1977) Maximum likelihood from incomplete data via the E-M algorithm (with discussion). Journal of the Royal Statistical Society, Ser. B., 39, 1-38.

[6] Diebolt, J. and Robert, C. P. (1994) Bayesian estimation of finite mixture distributions, Journal of the Royal Statistical Society, Series B, 56, 363-75

[7] Fraley, C. (1996) Algorithm for Model-Based Gaussian Hierarchical Clustering, Technical Report No.311, Department of Statistics, University of Washington

[8] Fraley, C. and Raftery, A.E. (1999) MCLUST: Software for Model-Based Cluster Analysis, Journal of Classification, 16, 297-306

[9] Fraley, C. and Raftery, A. E. (2000) How Many Cluster? Which Clustering Method? Answers via Model-Based Cluster Analysis, Computer Journal, 41, 578-788

[10] Fraley, C. and Raftery, A. E. (2002) Model-Based Clustering, Discriminate Analysis, and Density Estimation, Journal of the American Statistical Association, 97, 611-631

[11] Gelfand, A. E. and Smith, A. f. M. (1990) Sampling-Based Approaches to Calculating Marginal Densities, American Statistical Association, 85, 398-409.

[12] Geman, S. and Geman, D. (1984) Stochastic relaxation, Gibbs distribution and Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence, 6, 721–741

[13] Geyer J. (1992) Practical Markov Chain Monte Carlo, Statistical Science, Vol. 7, No. 4 , 473-483

[14] Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. M. (1998) Markov chain Monte Carlo in practice: a roundtable discussion. The American Statistician, 52, 93-100

[15] Kass, R. E. and Raftery, A. E. (1995) Bayes Factors, Journal of the American Statistical Association, 90, 773-795

[16] Kendall, Maurice George, Sir. (1975) Multivariate analysis. London; C. Griffin

[17] Hastie, T., Tibshirani, R. and Friedman J. (2001) The Elements of Statistical Learning, NY, Spring-Verlag

[18] Lewis, S. M. and Raftery, A. E. (1997) Estimating Bayes factors via posterior simulation with the Laplace-Metropolis estimator. Journal of the American Statistical Association, 92, 648-655

[19] McLachLan G. J., and Basford K. E. (1988) Mixture models: Inference and applications to clustering, John Wiley and Sons

[20] McLachlan G. J. and Krishnan J. (1996) The EM Algorithm and Extensions, NY, Wiley-Interscience

[21] McLachlan G. J. and Peel D. (2000) Finite Mixture models, Wiley Series in Probability and Statistics

[22] Muthén, B. and Shedden K. (1999) Finite mixture modeling with mixture outcomes using the EM algorithm. Biometrics, 55(2):463-469.

[23] Raftery, Adrian E. (1996) Hypothesis Testing and Model Selection Via Posterior Simulation. In Practical Markov Chain Monte Carlo (W.R.Gilks, D.J. Spiegelhalter and S. Richardson, Eds) London: Chapman and Hall, pp. 163-88

[24] Raftery, A.E. and Lewis, S.M. (1992) How many iterations in the Gibbs sampler?

In Bayesian Statistics 4 (J.M. Bernardo et al., editors), Oxford University Press, pp. 763-773.

[25] Raftery, A.E. and Lewis, S.M. The Number of Iterations, Convergence Diagnostics and Generic Metropolis Algorithm. In Practical Markov Chain Monte Carlo (W.R.Gilks, D.J. Spiegelhalter and S. Richardson, Eds) London: Chapman and Hall,

[26] Smith, F. M. and Roberts, G. O. (1993) Bayesian Computation via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods, Royal Statistical Society B, 55: 3-24.

[27] Spiegelhalter, D., A. Thomas, N. Best, and W. Gilks (1995a) Bayesian Inference Using Gibbs Sampling (BUGS), Ver. 0.5. MRC Biostatistics Unit, IPH, Cambridge, U.K.

[28] Srivastava, M.S. and Khatri, C.G. (1979) An Introduction to Multivariate Statistics, NY, North Holland.

[29] Tierney, L., and Kadane, J.B. (1986) Accurate Approximations for Posterior Moments and Marginal Densities, Journal of the American Statistical Association, 81, 82-86

[30] Yang, C. C. and Muthén B. (1997b) Finite mixture of generalized linear models using Gibbs sampling and E-M algorithm. American Statistical Association (ASA) Joint Statistics Meeting (JSM), Anaheim, CA, U.S.A.