The Performance of Expectation Maximization (EM) Algorithm in Gaussian Mixed Models (GMM)


2Department of Mathematics, Faculty of Science, Universiti Putra Malaysia, 43400 UPM Serdang, Selangor, Malaysia
3Faculty of Economics and Business, Universiti Kebangsaan Malaysia, 43600 UKM, Bangi, Selangor, Malaysia
*E-mail: izhan@tmrnd.com.my

ABSTRACT

Expectation Maximization (EM) algorithm has experienced a significant increase in terms of usage in many fields of study. In this paper, the performance of the said algorithm in finding the Maximum Likelihood for the Gaussian Mixed Models (GMM), a probabilistic model normally used in fraud detection and recognizing a person’s voice in speech recognition field, is shown and discussed. At the end of the paper, some suggestions for future research works will also be given.

Keywords: Expectation Maximization (EM), Gaussian Mixed Models (GMM), Box and Muller Transformation

INTRODUCTION

Every year, telecommunication companies register heavy loses due to fraud activities amounting to million of dollars. Vendors, seeing the above as an opportunity not to be missed, compete to provide data mining applications which could detect the said activity effectively using methods such as OLAP, deviation based outlier detection, Hidden Markov Model, and the model which became the focal area of this paper, the Gaussian Mixed Models (GMM).

GMM is best known in providing a robust speaker representation for the difficult task of speaker identification on short-time speech spectra, which is a cosine, transformed of log energy filter outputs from processed magnitude spectrum from a 20 ms short time segment of speech, by simulated me-scale filter-bank (Reynolds et al., 1995). Its function is further extended to detect fraud activities on daily number of calls and length of calls occurring during the office hours, the evening hours and the night hours for both national and international calls (Mohd Yusoff et al., 2006; Tanigushi et al., 1998).

Maximum likelihood estimation for GMM is difficult to find and the solution is Expectation Maximization (EM) algorithm. The EM algorithm was first introduced by Dempster et al. (1977) and since then, there has been a significant increase in terms of its usage, particularly in finding the
Maximum Likelihood for probabilistic models (such as missing data, grouping, censoring, truncation, and finite mixtures). The main issue with respect to the EM algorithm is finding the right choice of initial parameters and the number of components. This particular issue or problem is illustrated in several examples in this paper.

The subsequent sections provide a brief introduction of the EM algorithm and GMM, generate the simulation univariate and multivariate data with clear and hidden components, present the results gathered from the GMM and EM algorithm where the emphasis given on the choice of the initial parameters and the number of components, and some suggestions for future research works.

The Gaussian Mixed Models (GMM) and Expectation Maximization (EM) Algorithm

Let \( x \in \mathbb{R}^d \) and \( K \) be the number of components where each component having its own prior probability (weight) \( a_i \) and probability density function with the mean \( \mu_i \) and covariance \( \Sigma_i \). All of them are mixed resulting in the following formula, which is also known as the Gaussian Mixed Models (GMM):

\[
\sum_{i=1}^{K} a_i \phi(x | \mu_i, \Sigma_i) = \sum_{i=1}^{K} a_i \frac{1}{\sqrt{(2\pi)^d |\Sigma_i|}} \exp \left( -\frac{(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)}{2} \right)
\]

where prior probability (weight) of component \( i \) that is \( a_i \) satisfy the constraint \( \sum_{i=1}^{K} a_i = 1 \).

For the case of voice recognition, assuming there are \( n \) number of speakers. The \( m \) number of samples were collected from each speaker. Equation (1) is derived for each sample, where its parameters were kept in the database for comparison purposes. Fraud detection would follow similar steps.

From equation (1), the likelihood function and log likelihood function by

\[
L(X | \theta) = \prod_{j=1}^{n} f(x_j | \theta)
\]

and

\[
l(X | \theta) = \log L(X | \theta) = \sum_{j=1}^{n} \log \left( \sum_{i=1}^{K} a_i \phi(x_j | \mu_i, \Sigma_i) \right)
\]

were defined, respectively. The maximum likelihood estimation (m.l.e) aimed at finding \( \hat{\theta} \) which maximized \( l(X | \theta) \), with respect to \( \theta \) (Mardia et al., 1979). The expression \( \log \left( \sum_{i=1}^{K} a_i \phi(x_j | \mu_i, \Sigma_i) \right) \) in the log likelihood function is difficult to solve, and in order to overcome this problem, the Expectation Maximization (EM) algorithm was used.

In the EM algorithm, the distribution of \( X \) needs to be estimated in the sample space \( \chi \), but \( X \) can only be observed indirectly through \( Y \) in the sample space \( Y \). In many cases, there is a mapping \( x \rightarrow y(x) \) from \( \chi \circ Y \), and \( x \) is only known to lie in a subset of \( \chi \), denoted by \( \chi(y) \), which is determined by equation \( y = y(x) \). The distribution of \( X \) is parameterized by a family of distributions \( f(x | \theta) \), with parameters \( \theta \in \Omega \) or \( x \). The distribution of \( Y \), \( g(y | \theta) \) is therefore:

\[
g(y | \theta) = \int_{\chi(y)} f(x | \theta) dx
\]

The EM algorithm aims at finding \( \theta \) which maximizes \( g(y | \theta) \) given an observed \( y \). Let the function

\[
Q(\theta' | \theta) = E(\log f(x | \theta) | y, \theta)
\]
be the expected value of $\log f(x|\theta')$ given $y$ and $\theta$. The expectation was assumed to exist for all the pairs $(\theta',\theta)$. In particular, it was assumed that $f(x|\theta) > 0$ for $\theta \in \Omega$.

**EM Iteration**

E-Step: Compute $Q(\theta|\theta^n)$

M-step: Choose $\theta^{n+1}$ to be a value of $\theta \in \Omega$ that maximizes $Q(\theta|\theta^n)$ (Dempster et al., 1977). In the case of GMM, it was defined that $Q(\theta'|\theta) = E \log \prod_{i=1}^{n} a_{ij} \phi(x_i|\mu_j, \Sigma_j) | X, \theta^n$, where $y_i \in \{1,2,...,K\}$, $y_i=k$ if the $i^{th}$ sample was generated by the $k^{th}$ mixture component. It was simplified using (among other) the Bayes formula which is $f(\theta|x) \propto f(x|\theta)P(\theta)$, where $f(\theta|x) = \text{posterior probability}$, $f(x|\theta) = \text{likelihood function}$, and $P(\theta) = \text{prior probability}$ (Tsay, 2005; Bilmes, 1997) to the following equations:

$$ Q(\theta'|\theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} p_{ik} \log a_{ik} + \sum_{i=1}^{n} \sum_{k=1}^{K} p_{ik} \log \phi(x_i|\mu_k, \Sigma_k) $$

(4)

where

$$ p_{ik} = \frac{a_{ik} \phi(x_i|\mu_k, \Sigma_k)}{\sum_{j} a_{ij} \phi(x_i|\mu_j, \Sigma_j)} $$

(5)

and

$$ \phi(x_i|\mu_k, \Sigma_k) = \frac{1}{\sqrt{(2\pi)^d |\Sigma_k|}} \exp\left(-\frac{(x_i - \mu_k)'\Sigma_k^{-1}(x_i - \mu_k)}{2}\right) $$

(6)

**The EM Iteration (for GMM)**

E-Step:

Equation (5) is calculated.

M-Step:

The following formulas (derived from the Lagrange multipliers, $\frac{\partial Q}{\partial \mu_j} = 0$ and $\frac{\partial Q}{\partial \Sigma_j} = 0$, respectively) are calculated. Further details are given in Appendix A.2.

$$ a_{ij} = \frac{1}{n} \sum_{i} p_{ij} $$

(7)

$$ \mu_j = \frac{\sum_{i} p_{ij} x_i}{\sum_{i} p_{ij}} $$

(8)
\[
\Sigma_j = \frac{\sum_i p_{ij}(x_i - \mu_j)(x_i - \mu_j)^T}{\sum_i p_{ij}}
\]  

(9)

The above steps (i.e. E-step and M-step) were repeated until a convergence was achieved.

**SIMULATION DATA**

A program called “Simulate” was developed (using C++ language) to generate simulation data for equation (1) with the parameters as per given in Table 1 (taken from Everitt et al., 1981, with modifications) using Box and Muller Transformation (Box et al., 1958) and Equation (10). The simulation data were then labelled as “Sample1”, “Sample2”, “Sample3” and “Sample4”, and their histograms are shown in Figs. 1 and 2.

\[
z_j = \mu + (-2\sigma^2 \log u_j)^{1/2} \cos 2\pi u_{j+1}
\]

(10)

\[
z_{j+1} = \mu + (-2\sigma^2 \log u_j)^{1/2} \sin 2\pi u_{j+1}, u_j, u_{j+1} \sim U(0,1)
\]

The “Simulate” program would generate one random number, denoted by \(U_1\), from the uniform distribution \(U(0,1)\), and check whether it was less than say \(a_i\) \((i=1,2)\). If the answer is “yes”, the two random numbers, denoted by \(U_1\) and \(U_2\), are generated from the uniform distribution \(U(0,1)\) and used in the computing equation (10), along with the corresponding \(\mu_i\) and \(\sigma_i^2\), taken from Table 1. In this study, these steps were repeated until 1000 observations were obtained. For “Sample 4”, apart from equation (10), the formulas given in Appendix A.1 (in the matrix format) were also used.

In Fig. 1.1, two humps are observed and these represent two components: \((\mu_1, \sigma_1) = (0.0, 1.0)\) and \((\mu_2, \sigma_2) = (2.0, 0.25)\). Both of them are well-separated, in which the observations for the latter component are grouped around the mean.

One would never expect to find the two components in Fig. 1.2. The histogram is dominated by the component \((\mu_1, \sigma_1) = (0.0, 1.0)\) due to the fact that \(a_1 = 0.85\).

In Fig. 1.3, two humps are vividly displayed and they represent two components: \((\mu_1, \sigma_1) = (-1.0, 0.25)\) and \((\mu_2, \sigma_2) = (4.0, 4.0)\). The third component, \((\mu_3, \sigma_3) = (0.0, 1.0)\), is hidden from the view by the two components indicated earlier. The observations are grouped around the mean for the component \((\mu_2, \sigma_2) = (-1.0, 0.25)\).

The histograms in Figs. 2.1 and 2.2 appear to split into two representing components \((\mu_1, \mu_2) = (5.01, 5.91)\) and \((\mu_1, \mu_2) = (2.78, 2.95)\), respectively; whereas Figs. 2.3 and 2.4 into three representing components \((\mu_1, \mu_3, \mu_4) = (1.46, 4.2, 5.48)\) and \((\mu_1, \mu_3, \mu_4) = (0.25, 1.3, 1.98)\), respectively.
The Performance of Expectation Maximization (EM) Algorithm in Gaussian Mixed Models (GMM)

TABLE 1

\(a\)'s, \(\mu\)'s, \(\sigma\)'s and \(\Sigma\)'s for each sample used in the “Simulate” program. The number of observations generated by the program is given in the bracket

<table>
<thead>
<tr>
<th>Sample</th>
<th>(a_1)</th>
<th>(\mu_1)</th>
<th>(\sigma_{11})</th>
<th>(\Sigma_1)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sample 1</td>
<td>0.4</td>
<td>0.0</td>
<td>1.0</td>
<td>0.12 0.01</td>
</tr>
<tr>
<td>(N=1000)</td>
<td></td>
<td>2.0</td>
<td>0.02 0.01</td>
<td>0.14 0.01</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.25 0.01</td>
<td>0.03 0.01</td>
</tr>
</tbody>
</table>

| Sample 2 | 0.85    | 0.0       | 1.0             | 0.27 0.09   |
| (N=1000) |         | 2.0       | 0.18 0.09       | 0.09 0.04   |
|          |         |           | 0.25 0.02       | 0.2 0.06    |

| Sample 3 | 0.33    | 0.0       | 1.0             | 0.38 0.09   |
| (N=1000) |         | -1.0      | 0.3 0.03        | 0.11 0.08   |
|          |         | 4.0       | 0.06 0.05       | 0.32 0.07   |

| Sample 4 | 0.33    |           |                 |             |
| (N=1000) |         |           |                 |             |
|          |         |           |                 |             |
Fig. 1: The histograms of “Sample1” (with overall mean and standard deviation equal to 1.28 and 1.19, respectively); “Sample2” (with overall mean and standard deviation equal to 0.34 and 1.19, respectively); and “Sample3” (with overall mean and standard deviation equal to 1.18 and 2.62, respectively)
The Performance of Expectation Maximization (EM) Algorithm in Gaussian Mixed Models (GMM)

RESULTS

A program known as the “GMM” was developed using the Java language to find the parameters of equation (1) by employing the EM algorithm, where iteration is stopped when $|\theta^{p+1} - \theta^p| < 0.000001$. Other methods involved in the calculation of EM algorithm include the Cholesky method (Mardia et al., 1979). In this section, two scenarios are therefore presented.

Scenario 1: In Table 2, with the exception of “Sample4” (where initial parameters were taken from Everitt et al., 1981), the initial parameters for “Sample1”, “Sample2” and “Sample3” were determined using visual inspection of the histograms given in Fig. 1. This was done by concentrating on the observation(s) that gave the highest frequency, as shown by the components which were clearly displayed.

Fig. 2: The histograms of “Sample4” $x_1$ (with overall mean and standard deviation equal to 5.88 and 0.81, respectively); $x_2$ (with overall mean and standard deviation equal to 3.05 and 0.44, respectively); $x_3$ (with overall mean and standard deviation equal to 3.79 and 1.75, respectively); and $x_4$ (with overall mean and standard deviation equal to 1.2 and 0.82, respectively)
Mohd. Izhan Mohd. Yusoff, Mohd. Rizam Abu Bakar and Abu Hassan Shaari Mohd Nor

TABLE 2

a’s, μ’s, σ’s and Σ’s for each sample used in the GMM program,
where they were treated as the initial parameters

<table>
<thead>
<tr>
<th>Sample</th>
<th>a₁</th>
<th>μ₁</th>
<th>σ₁₁</th>
<th>a₂</th>
<th>μ₂</th>
<th>σ₂₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>“Sample1”</td>
<td>0.5</td>
<td>0.0</td>
<td>1.0</td>
<td>0.5</td>
<td>2.0</td>
<td>1.0</td>
</tr>
<tr>
<td>“Sample2”</td>
<td>0.5</td>
<td>0.0</td>
<td>1.0</td>
<td>0.5</td>
<td>1.5</td>
<td>1.0</td>
</tr>
<tr>
<td>“Sample3”</td>
<td>0.33</td>
<td>0.0</td>
<td>1.0</td>
<td>0.33</td>
<td>-1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>“Sample4”</td>
<td>0.33</td>
<td>0.34</td>
<td>1.0</td>
<td>0.33</td>
<td>4.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

The values given in Table 2 were used by the “GMM” program as the initial parameters to find the final ones for the simulation data, as shown in Figs. 1 and 2. The results are as tabulated below.
The Performance of Expectation Maximization (EM) Algorithm in Gaussian Mixed Models (GMM)

TABLE 3
\( \hat{\theta}, \hat{\mu}, \hat{\sigma}, \hat{\Sigma} \) for each sample produced by the “GMM” program, using (Table 2) as the initial parameters. The “GMM” program converged is given in the bracket.

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Sample 1</th>
<th>Sample 2</th>
<th>Sample 3</th>
<th>Sample 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(Converged at iteration no 156)</td>
<td>(Converged at iteration no 389)</td>
<td>(Converged at iteration no 316)</td>
<td>(Converged at iteration no 32)</td>
</tr>
<tr>
<td>( \hat{\theta} )</td>
<td>( \hat{\mu} )</td>
<td>( \hat{\sigma} )</td>
<td>( \hat{\Sigma} )</td>
<td>( \hat{\theta} )</td>
</tr>
<tr>
<td>( \hat{\alpha}_1 )</td>
<td>( \hat{\alpha}_2 )</td>
<td>( \hat{\mu}_1 )</td>
<td>( \hat{\mu}_2 )</td>
<td>( \hat{\sigma}_{11} )</td>
</tr>
<tr>
<td>0.34</td>
<td>0.66</td>
<td>-0.05</td>
<td>1.98</td>
<td>0.85</td>
</tr>
<tr>
<td>0.82</td>
<td>0.18</td>
<td>-0.02</td>
<td>1.97</td>
<td>0.93</td>
</tr>
<tr>
<td>0.29</td>
<td>0.31</td>
<td>-0.1</td>
<td>1.01</td>
<td>3.76</td>
</tr>
<tr>
<td>0.43</td>
<td>1.45</td>
<td>0.23</td>
<td>( \hat{\Sigma}_1 )</td>
<td>0.15</td>
</tr>
<tr>
<td>5.93</td>
<td>2.77</td>
<td>4.22</td>
<td>1.30</td>
<td>( \hat{\Sigma}_2 )</td>
</tr>
<tr>
<td>6.58</td>
<td>2.94</td>
<td>5.53</td>
<td>1.99</td>
<td>( \hat{\Sigma}_3 )</td>
</tr>
</tbody>
</table>

It is crucial to note that for the univariate samples, the convergence was achieved with more than 100 iterations, while for the multivariate samples, less than 100 iterations were required. The choice of the initial parameters might play an important role in making the convergence process faster, as illustrated by the latter.

The “GMM” program managed to find (final) parameters even in cases where the components were hidden from the view, but this is provided that the number of components and the observations which give the highest frequency for the identifiable components are known.

Scenario 2: Great care should be taken when choosing the initial parameters (to start the EM algorithm) as well as the number of components, where wrong choice will lead to the situation exemplified in Table 4. Other examples can be found in Everitt et al. (1981) and Reynolds et al. (1995).
The initial (1st row) and final (2nd row) parameters for “Sample 3” (chosen for having hidden components), where: two components were used for (4.1), four components were used for (4.2), and six components were used for (4.3) and (4.4). The actual number of the components is three.

<table>
<thead>
<tr>
<th>“Sample 3”</th>
<th>(a_1=0.5)</th>
<th>(\mu_1=-1.0)</th>
<th>(\sigma_1=1.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_2=0.5)</td>
<td>(\mu_2=4.0)</td>
<td>(\sigma_2=1.0)</td>
<td></td>
</tr>
<tr>
<td>“Sample 3”</td>
<td>(\hat{a}_1=0.51)</td>
<td>(\hat{\mu}_1=-0.75)</td>
<td>(\hat{\sigma}_1=0.45)</td>
</tr>
<tr>
<td>(Converged at iteration no 99)</td>
<td>(\hat{a}_2=0.49)</td>
<td>(\hat{\mu}_2=3.21)</td>
<td>(\hat{\sigma}_2=5.58)</td>
</tr>
</tbody>
</table>

(4.1)

<table>
<thead>
<tr>
<th>“Sample 3”</th>
<th>(a_1=0.25)</th>
<th>(\mu_1=-1.0)</th>
<th>(\sigma_1=1.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_2=0.25)</td>
<td>(\mu_2=4.0)</td>
<td>(\sigma_2=1.0)</td>
<td></td>
</tr>
<tr>
<td>(a_3=0.25)</td>
<td>(\mu_3=0.0)</td>
<td>(\sigma_3=1.0)</td>
<td></td>
</tr>
<tr>
<td>(a_4=0.25)</td>
<td>(\mu_4=0.0)</td>
<td>(\sigma_4=1.0)</td>
<td></td>
</tr>
<tr>
<td>“Sample 3”</td>
<td>(\hat{a}_1=0.31)</td>
<td>(\hat{\mu}_1=-1.01)</td>
<td>(\hat{\sigma}_1=0.22)</td>
</tr>
<tr>
<td>(Converged at iteration no 186)</td>
<td>(\hat{a}_2=0.41)</td>
<td>(\hat{\mu}_2=3.76)</td>
<td>(\hat{\sigma}_2=4.71)</td>
</tr>
<tr>
<td>(\hat{a}_3=0.14)</td>
<td>(\hat{\mu}_3=0.1)</td>
<td>(\hat{\sigma}_3=0.89)</td>
<td></td>
</tr>
<tr>
<td>(\hat{a}_4=0.14)</td>
<td>(\hat{\mu}_4=0.1)</td>
<td>(\hat{\sigma}_4=0.89)</td>
<td></td>
</tr>
</tbody>
</table>

(4.2)

<table>
<thead>
<tr>
<th>“Sample 3”</th>
<th>(a_1=0.17)</th>
<th>(\mu_1=-1.0)</th>
<th>(\sigma_1=1.0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a_2=0.17)</td>
<td>(\mu_2=4.0)</td>
<td>(\sigma_2=1.0)</td>
<td></td>
</tr>
<tr>
<td>(a_3=0.17)</td>
<td>(\mu_3=0.0)</td>
<td>(\sigma_3=1.0)</td>
<td></td>
</tr>
<tr>
<td>(a_4=0.17)</td>
<td>(\mu_4=0.0)</td>
<td>(\sigma_4=1.0)</td>
<td></td>
</tr>
<tr>
<td>(a_5=0.17)</td>
<td>(\mu_5=0.0)</td>
<td>(\sigma_5=1.0)</td>
<td></td>
</tr>
<tr>
<td>(a_6=0.17)</td>
<td>(\mu_6=0.0)</td>
<td>(\sigma_6=1.0)</td>
<td></td>
</tr>
<tr>
<td>“Sample 3”</td>
<td>(\hat{a}_1=0.31)</td>
<td>(\hat{\mu}_1=-1.01)</td>
<td>(\hat{\sigma}_1=0.22)</td>
</tr>
<tr>
<td>(Converged at iteration no 312)</td>
<td>(\hat{a}_2=0.41)</td>
<td>(\hat{\mu}_2=3.76)</td>
<td>(\hat{\sigma}_2=4.71)</td>
</tr>
<tr>
<td>(\hat{a}_3=0.07)</td>
<td>(\hat{\mu}_3=0.1)</td>
<td>(\hat{\sigma}_3=0.89)</td>
<td></td>
</tr>
<tr>
<td>(\hat{a}_4=0.07)</td>
<td>(\hat{\mu}_4=0.1)</td>
<td>(\hat{\sigma}_4=0.89)</td>
<td></td>
</tr>
<tr>
<td>(\hat{a}_5=0.07)</td>
<td>(\hat{\mu}_5=0.1)</td>
<td>(\hat{\sigma}_5=0.89)</td>
<td></td>
</tr>
<tr>
<td>(\hat{a}_6=0.07)</td>
<td>(\hat{\mu}_6=0.1)</td>
<td>(\hat{\sigma}_6=0.89)</td>
<td></td>
</tr>
</tbody>
</table>

(4.3)
The Performance of Expectation Maximization (EM) Algorithm in Gaussian Mixed Models (GMM)

Notice that Table 4.2's \( n_3 = n_4 = -0.1 \) and \( v_{33} = v_{44} = 0.89 \) and if \( a_3 + a_4 \) were computed, 0.28 would therefore be obtained, and this is no far different from the ones given in Table 3. Table 4.3 also shows similar results, where \( \hat{\mu}_3 = \ldots = \hat{\mu}_6 = -0.1, = \hat{\sigma}_{33} = \ldots = \hat{\sigma}_{66} = 0.89 \) and \( \hat{a}_3 = \ldots = \hat{a}_6 = 0.07 \) where \( \hat{a}_3 + \ldots + \hat{a}_6 = 0.28 \). Despite converging at iteration no. 2 (the lowest so far), the final parameters shown in Table 4.4 are completely different from those in Table 3, and this is a direct consequence from ignoring the characteristics shown by the observations in the histograms.

CONCLUSIONS

In the previous sections, “Sample1”, “Sample2”, “Sample3”, and “Sample4” (using a program called “Simulate”) were generated with known number of both components and parameters. Using the same information, particularly on the number of components and determining the initial parameters to start the EM algorithm by inspecting the histograms, the final parameters produced from the EM algorithm (using the program known as the “GMM”) are similar to the real ones.

Just to show how important the process of choosing the initial parameters is (to start the EM algorithm) and the number of components, “Sample3” was selected for having hidden components, while the process of determining the initial parameters to start EM algorithm (i.e. by inspecting the histograms) and reducing the number of components was repeated; the final parameters produced were incorrect. The same results were also obtained when the number of components was increased; for the initial parameters to start the EM algorithm, let the mean equals to 0 and the standard deviation equals to 1 (a common mistake done by most of the practitioners).

In contrary to the above, when the number of components was increased and the initial parameters to start the EM algorithm was determined by inspecting the histograms and for the rest (especially the hidden components) by letting the mean equals to 0 and standard deviation equals to 1, the final parameters produced (with minor adjustments) were similar to the real ones (a “characteristic” where some might consider it as unimportant and therefore choose to ignore).

The determination of the initial parameters to start the EM algorithm could be made easier and faster using the graphical techniques such as plotting \( \log \frac{\phi_{i+1}}{\phi_i} \) against \( x \), where each approximately straight line, with negative slope represents an area where one component dominates and the kernel...
method defined by \( f(\tilde{t}_i) = \sum_{k=1}^{m} \exp\left(-\frac{2\sigma^2}{m}\right) \exp\left(-\frac{1}{2r^2} \left(\frac{2\sigma^2}{b-a}\right)\right) \sum_{k=1}^{m} \exp\left(\frac{2\pi k l}{m}\right), \quad m = 2^r \) (Everitt et al., 1981; Bhattacharya, 1967; Silverman, 1986). Nevertheless, the main disadvantage of both methods is that they can not be used to detect hidden components.

Appendix A

A.1 “Simulate” program uses the following formulas to produce “Sample4” (where the subscript represents the dimension of the matrix).

\[
X_{\text{nt}} = C_{\text{nxn}} Z_{\text{nx1}} + \mu_{\text{nx1}}, \quad \Sigma_{\text{nxn}} = C_{\text{nxn}} C_{\text{nxn}}^t
\]

where \( C_{ij} = \sigma_{ij} - \sum_{k=1}^{j-1} c_{ik} c_{jk} \), \( j \leq i \), and \( z_i \), \( i \)th component of \( Z \), is as per defined in equation (10), where \( \mu \) and \( \sigma \) are set/fixed at 0 and 1, respectively.

A.2 Derivation of Equations (7), (8) and (9)

A.2.1 Using Lagrange multipliers defined by \( \max/min F(x, y, z) \) subject to \( \Phi(x, y, z) = 0 \), \( G(x, y, z) = F(x, y, z) + \lambda \Phi(x, y, z) \), \( \frac{\partial G}{\partial x} = 0, \frac{\partial G}{\partial y} = 0, \frac{\partial G}{\partial z} = 0 \) (Spiegel, 1974) on \( \max \sum \sum \rho_{ij} \log(a_i) \) subject to \( \sum_j a_j = 1 \) (or \( \sum_j a_j - 1 = 0 \)). Equation (7) would be obtained.

A.2.2 From \( \frac{\partial}{\partial \rho_{ij}} \left( \frac{1}{2} \sum_j \sum_i \rho_{ij} (x_i - \mu_i)\Sigma_j^{-1} (x_i - \mu_i) \right) = 0 \), equation (8) would be obtained using the following matrix properties, \( \frac{\partial x^t A y}{\partial x} = A y, \frac{\partial a^t x}{\partial x} = a \).

A.2.3 The first and second expressions of

\[
\frac{\partial}{\partial \Sigma_j} \left( \frac{1}{2} \sum_j \sum_i \rho_{ij} (x_i - \mu_i)\Sigma_j^{-1} (x_i - \mu_i) \right) + \frac{\partial}{\partial \Sigma_j^{-1}} \left( \frac{1}{2} \sum_i \sum_j \rho_{ij} \log |\Sigma_j^{-1}| \right) = 0 \]

use the following matrix properties, \( \frac{\partial \text{tr}(xy)}{\partial x} = y + y^t - \text{Diag}(y) \), and \( \sum x_i A x_i = \text{tr}(A \sum x_i x_i^t) \) to get equation (9). (Mardia et al., 1979).
REFERENCES


