A Quick Guide for the EMCluster Package

Wei-Chen Chen\textsuperscript{1}, Ranjan Maitra\textsuperscript{2}

\textsuperscript{1}pbdR Core Team

\textsuperscript{2}Department of Statistics,
Iowa State University,
Ames, IA, USA

Contents

Acknowledgement \hspace{1cm} ii
1. Introduction \hspace{1cm} 1

2. EM Algorithm and Initialization \hspace{1cm} 2

3. Examples \hspace{1cm} 3
3.1. EM Control \hspace{1cm} 3
3.2. Unsupervised Clustering \hspace{1cm} 3
3.3. Semi-supervised Clustering \hspace{1cm} 4

References \hspace{1cm} 6
Acknowledgement

Chen was supported in part by the project “Bayesian Assessment of Safety Profiles for Pregnant Women From Animal Study to Human Clinical Trial” funded by U.S. Food and Drug Administration, Office of Women’s Health. The project was supported in part by an appointment to the Research Participation Program at the Center For Biologics Evaluation and Research administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and the U.S. Food and Drug Administration.

Chen was supported in part by the Department of Ecology and Evolutionary Biology at the University of Tennessee, Knoxville, and a grant from the National Science Foundation (MCB-1120370.)

Chen was supported in part by the project “Visual Data Exploration and Analysis of Ultra-large Climate Data” funded by U.S. DOE Office of Science under Contract No. DE-AC05-00OR22725.

The authors acknowledge partial support by the National Science Foundation CAREER Grant # DMS-0437555.
1. Introduction

We use this section to show how to install EMCluster, introduce basic finite mixture Gaussian distribution, and notation for the corresponding variables and objects. In Section 2, we introduce EM algorithms and strategies of initialization for model-based clustering, and the major R functions are also introduced. In Section 3, we provide two examples for unsupervised and semi-supervised clustering, and quick demos are shown.

1.1. Installation

The EMCluster has simple interface of R (R Core Team 2012) to efficient C code that we optimize the algorithm and utilize LAPACK library for matrix algebra. The package should install on most popular platform without further configurations. The installation can be done in the shell command as

Shell Command

```bash
> R CMD INSTALL EMCluster_0.2-0.tar.gz
```

or from any R session as

Shell Command

```r
R> install.packages("EMCluster")
```

with user-favored CRAN mirror site.

1.2. Notation

The EMCluster assumes finite mixture Gaussian distribution with unstructured dispersion and implements EM algorithm for model-based clustering in both unsupervised and semi-supervised clustering. The model is

\[ f(x|\vartheta) = \sum_{k=1}^{K} \pi_k \phi(x|\mu_k, \Sigma_k). \]  

- \( x \) is a \( p \)-dimensional observation.
- \( \vartheta = \{ \pi_1, \pi_2, \ldots, \pi_{K-1}, \mu_1, \mu_2, \ldots, \mu_K, \Sigma_1, \Sigma_2, \ldots, \Sigma_K \} \).
- \( \pi_1, \pi_2, \ldots, \pi_K \) are mixing proportion for \( K \) components that \( \sum_{k=1}^{K} \pi_k = 1 \) and \( 0 < \pi_k < 1 \) for all \( k = 1, 2, \ldots, K \).
• $\phi(x|\mu_k, \Sigma_k)$'s are multivariate Gaussian densities with mean $\mu_k$ and dispersion $\Sigma_k$.

• $\mu_1, \mu_2, \ldots, \mu_K$ are $p$-dimensional mean vectors and $\Sigma_1, \Sigma_2, \ldots, \Sigma_K$ are $p \times p$-dimensional dispersion matrices.

We also assume the following notation for R objects in EMCluster

• $x$ is a matrix with dimension $n \times p$ containing $n$ observations to be clustered.

• $\pi$ is a vector with length $K$ containing $\pi_1, \pi_2, \ldots, \pi_K$.

• $\mu$ is a matrix with dimension $K \times p$ containing $\mu_1, \mu_2, \ldots, \mu_K$.

• $\Sigma$ is a matrix with dimension $K \times p(p+1)/2$ containing low triangular matrices of $\Sigma_1, \Sigma_2, \ldots, \Sigma_K$.

• $\lambda$ is a vector with length $n$ containing labels of observations. If $\lambda=$NULL, then unsupervised clustering is performed, otherwise semi-supervised clustering is performed where labels can be $1, 2, \ldots, K$ for cluster-known data and 0 for cluster-unknown data.

2. EM Algorithm and Initialization

For $n$ observations $X = \{x_1, x_2, \ldots, x_n\}$, the log likelihood based on the Equation (1) is

$$\log L(\theta|X) = \log \left( \prod_{i=1}^{n} f(x_i|\theta) \right).$$

A standard way to optimize the Equation (2) is to utilize the EM algorithm (Dempster et al. 1977) composed of expectation (E-) and maximization (M-) steps.

The R function emcluster implements the algorithm to find the maximum likelihood estimates (MLEs) $\hat{\theta}$. The R functions e.step and m.step implements the both steps, respectively, which are useful for advanced developers.

Model-based clustering partition data into $K$ clusters by the maximum posterior

$$\arg\max_k \pi_k \phi(x|\mu_k, \Sigma_k)$$

for each observation. The R function assign.class provides this procedure returning cluster id $(1, 2, \ldots, K)$ for each $x_i$.

The initialization is an solution of avoiding local optimization when applying the EM algorithm to cluster high-dimensional data. The local optimization problem is more serious when data contain high overlaps of clusters. There are three initialization methods implemented in EMCluster: RndEM (Maitra 2009), emEM (Biernacki et al. 2003), and svd (Maitra 2001).

The R functions are

• init.EM calls rand.EM and em.EM and implements for RndEM and emEM methods, respectively, and

• emgroup implements the svd method.
The short summary of the initialization methods is

- **RndEM** repeats \( .\text{EMC}$.\text{short.iter} \) times of randomly selecting \( K \) centers from data and group all other data to the closest center based on Euclidean distance. Pick the best initial in terms of the highest log likelihood. Then, start EM algorithm from the best initial until convergence.

- **emEM** is composed of short-EM and long-EM steps. The short-EM step starts with randomly selecting \( K \) centers and applies EM iterations until convergence with loose tolerance \( .\text{EMC}.\text{short.eps} \). Repeat a few time until the total EM iterations reach \( .\text{EMC}.\text{short.iter} \). Then, start long-EM algorithm from the best initial until convergence with tight tolerance \( .\text{EMC}.\text{EM.eps} \).

- **svd** utilizes singular value decomposition to decomposition data, and select centers from the major component space determining by the singular values. The data are grouped to the centers by kmeans to generate an initial. Then, start EM algorithm from the initial until convergence.

3. Examples

3.1. EM Control

The **EMCluster** provides a control function \( .\text{EMControl} \) and three default controls \( .\text{EMC}, .\text{EMC}.\text{Rnd}, \) and \( .\text{EMC}.\text{Rndp} \) which are useful to manipulate different initializations. By default \( .\text{EMC} \) is for \( \text{emEM} \), \( .\text{EMC}.\text{Rnd} \) is for \( \text{RndEM} \), while \( .\text{EMC}.\text{Rndp} \) is for \( \text{RndEM} \) but with different flavor.

For larger data or high overlaps data, general users may want to increase the initial iteration \( ( .\text{EMC}.\text{short.iter} ) \) which can improve the clustering results. However, the initialization also cost computing time and may take longer to obtain stable initials especially for large number of clusters or high overlap data.

Anyway, these also provide advanced developers to explore new initialization methods. We give two examples next for unsupervised and semi-supervised clusterings utilizing all initialization methods in **EMCluster**.

3.2. Unsupervised Clustering

In an R session, you can run the demo as

```r
R> demo(allinit, 'EMCluster', ask = F)
```

which clusters a small dataset \( \text{da1} \) of **EMCluster** with 10 clusters. The data set has only 500 observations in two dimensions.

This demo provides a plot with four unsupervised clustering results including \( \text{emEM} \) (em), \( \text{RndEM} \) (Rnd), \( \text{RndEM}+ \) (Rnd+) and \( \text{svd} \) (svd) where colors and symbols indicate different clusters as in the Figure 1. The \( \text{RndEM}+ \) has different settings to the \( \text{RndEM} \). The demo also
compares classifications with true ids using adjusted Rand index (adjR) which has value before 0 (bad results) and 1 (perfect match) as in the Table 1.

Figure 1: Unsupervised clustering results for four initializations.

<table>
<thead>
<tr>
<th></th>
<th>logL</th>
<th>adjR</th>
</tr>
</thead>
<tbody>
<tr>
<td>em</td>
<td>-5657.603</td>
<td>0.9929422</td>
</tr>
<tr>
<td>Rnd</td>
<td>-5658.984</td>
<td>0.9096019</td>
</tr>
<tr>
<td>Rnd+</td>
<td>-5657.612</td>
<td>0.9929422</td>
</tr>
<tr>
<td>svd</td>
<td>-5657.600</td>
<td>0.9929422</td>
</tr>
</tbody>
</table>

Table 1: Comparison by log likelihood (logL) and adjusted Rand index (adjR) of four initializations.

3.3. Semi-supervised Clustering

Inside an R session, you can run the demo as

```R
R> demo(allinit_ss, 'EMCluster', ask = F)
```

which clusters a small dataset da1 in EMCluster with 10 clusters.

We randomly choose 6 clusters, then select 50% of data for each cluster. There are about 170 points are labeled with true which are prior information, then we cluster the rest 330 points into 10 clusters by semi-supervised clustering.
This demo provides a plot with three semi-supervised clustering results including \textit{emEM} (em), \textit{RndEM} (Rnd), \textit{RndEM+} (Rnd+) where colors and symbols indicate different clusters as in the Figure 2. Note that \textit{svd} method is not implemented for semi-supervised clustering. The demo also compares classifications with true ids using adjusted Rand index as in the Table 2. Comparing with the Table 1, the semi-supervised clustering provides better results than the unsupervised clustering in terms of higher log likelihood (logL) and adjusted Rand index (adjR). The reason is that some tiny clusters are not seeded at the initial step in unsupervised clustering, therefore, a few prior information from there can substantially improve accuracy by the semi-supervised clustering.

**Figure 2**: Semi-supervised clustering results for three initializations.

<table>
<thead>
<tr>
<th></th>
<th>logL</th>
<th>adjR</th>
</tr>
</thead>
<tbody>
<tr>
<td>em</td>
<td>-5657.882</td>
<td>0.9938959</td>
</tr>
<tr>
<td>Rnd</td>
<td>-5657.880</td>
<td>0.9938959</td>
</tr>
<tr>
<td>Rnd+</td>
<td>-5657.881</td>
<td>0.9938959</td>
</tr>
</tbody>
</table>

Table 2: Comparison by log likelihood (logL) and adjusted Rand index (adjR) of three initializations.
References


Affiliation:

Wei-Chen Chen
E-mail: wccsnow@gmail.com
URL: http://snoweye.github.io/