Penalized Model-Based Clustering

by

Wei Zhang

Advisor:

Dr. Kang Ling James

Department of Mathematics and Statistics
University of Minnesota Duluth, Duluth, MN 55812

June 2009
# Contents

Abstract ............................................................................................................................................. 1

Chapter 1 Introduction ...................................................................................................................... 2

Chapter 2 Models ............................................................................................................................. 3

2.1 Standard model-based clustering ................................................................................................ 3
  2.1.1 Model assumptions .............................................................................................................. 3
  2.1.2 Estimation via EM algorithm ............................................................................................ 4

2.2 Penalized model-based clustering ............................................................................................. 7
  2.2.1 Penalizing mean parameters with an $L_1$ penalty ......................................................... 7
  2.2.2 Penalizing mean parameters with an $L_\rho$ penalty ...................................................... 11

2.3 Model selection ......................................................................................................................... 14

2.4 Independent assessment of clusters ......................................................................................... 15

Chapter 3 Examples ....................................................................................................................... 17

3.1 CNS tumors data ...................................................................................................................... 17

3.2 Novartis multi-tissue data ......................................................................................................... 18

Chapter 4 Discussion ..................................................................................................................... 20

Acknowledgements ....................................................................................................................... 21

References ....................................................................................................................................... 22

Appendix .......................................................................................................................................... 23
Abstract

Clustering is the identification of groups of observations that are cohesive and separated from other groups. Most clustering done in practice is based largely on heuristic but intuitively reasonable procedures, such as hierarchical agglomerative clustering and K-means clustering. However, the statistical properties of these methods are generally unknown, precluding the possibility of formal inference, i.e. it seems difficult to define what is meant by the ‘right’ number of clusters. Clustering algorithms based on probability models offer a principled alternative to heuristic-based algorithms. In particular, the model-based clustering assumes that the data are generated by a finite mixture of underlying distributions, such as multivariate normal distribution for continuous case, and Poisson for discrete case. With the underlying probability model, the problem of determining the number of clusters becomes a statistical model choice problem.

This paper reviews the standard model-based clustering and penalized model-based approach with an $L_1$ penalty, and extends the current penalized method to the one with an $L_p$ penalty while $0 < p < 1$. It is shown theoretically that there is no clear shrinkage for the extended method. Therefore, we only illustrate the potential usefulness of the standard approach and the penalized one with an $L_1$ penalty by applying both methods to real gene expression data, and comparing the performances of these two clustering methods.

Keywords: mixture models, EM, penalized likelihood, clustering
Chapter 1

Introduction

Clustering is the identification of groups of observations that are cohesive and separated from other groups. Most clustering done in practice is based largely on heuristic but intuitively reasonable procedures, such as hierarchical agglomerative clustering and K-means clustering. However, the statistical properties of these methods are generally unknown, precluding the possibility of formal inference, i.e. it seems difficult to define what is meant by the ‘right’ number of clusters. Clustering algorithms based on probability models offer a principled alternative to heuristic-based algorithms. In particular, model-based clustering assumes that the data are generated by a finite mixture of underlying distributions, such as multivariate normal distribution for the continuous case, and Poisson for the discrete case. With the underlying probability model, the problem of determining the number of clusters becomes a statistical model choice problem (Fraley and Raftery, 1998).

Another crucial problem in cluster analysis is how to choose informative variables, which is more challenging in “high dimension, low sample size” settings, for example, in clustering gene expression data. Pan and Shen (2007) proposed a penalized model-based approach with an $L_1$ penalty function, in the context of mixtures of multivariate normal models with a common diagonal covariance matrix across all clusters, automatically realizing variable selection and parameter estimation.

This paper reviews the standard model-based clustering and penalized model-based approach with an $L_1$ penalty, and extends the current penalized method to the one with an $L_p$ penalty while $0 < p < 1$. It is shown theoretically that there is no clear shrinkage for the extended method. Therefore, we only illustrate the potential usefulness of the standard approach and the penalized one with an $L_1$ penalty by applying both methods to real gene expression data, and comparing the performances of these two clustering methods. The rest of this article is organized as follows. Section 2 reviews the standard model-based clustering and the penalized one with an $L_1$ penalty, and extends it to the penalized clustering with an $L_p$ penalty. The detailed EM algorithms are derived to compute the maximum likelihood estimates for all these three methods. A model selection criterion and an objective criterion for the evaluation of clustering approaches are given. Section 3 shows the results of the first two clustering methods when applied to two gene expression data. Section 4 presents a summary and a short discussion.
Chapter 2

Models

Given a set of $K$-dimensional data $X = \{x_j : j = 1, 2, ..., n\}$, where $x_j = \{x_{jk} : k = 1, 2, ..., K\}$ and $x_{jk}$ is the measurement of the $k$th attribute, the goal of clustering is to partition the observed data of previously unknown structure into a set of exhaustive and non-overlapping clusters so that objects in the same cluster are more similar to each other than objects from different clusters.

2.1 Standard model-based clustering

2.1.1 Model assumptions

Finite mixtures of distributions provide a flexible as well as rigorous approach to modeling various random phenomena. In model-based clustering, each cluster is mathematically represented by a parametric distribution, such as Gaussian for continuous case. Therefore, for gene expression data, the use of Gaussian components in the mixture distribution is natural. With a Gaussian mixture model-based clustering, it is assumed that each observation $x_j$ is taken to be a realization from a multivariate Gaussian mixture distribution with the probability density function

$$f(x_j; \Theta) = \sum_{i=1}^{g} \pi_i f_i(x_j; \theta_i),$$

where $\pi_i$'s are mixing proportions and $f_i(x_j; \theta_i)$ denotes the multivariate normal density function corresponding to cluster $i$ with its parameters $\mu_i$ (mean vector) and $\Sigma_i$ (covariance matrix). $\Theta$ represents all unknown parameters $\{(\pi_i, \theta_i) : i = 1, 2, ..., g\}$ in a $g$-component mixture model, with restrictions $0 \leq \pi_i \leq 1$ and $\sum_{i=1}^{g} \pi_i = 1$.

Data generated by mixtures of multivariate normal densities are characterized by clusters centered at the mean $\mu_i$. The corresponding surfaces of constant density are ellipsoidal. The structure of the covariance matrix $\Sigma_i$ determines the geometric features (shape, volume and orientation) of the clusters. For particular problems, some cross-cluster constraints may be imposed. Common instances include $\Sigma_i = \lambda I$, where all clusters are spherical and of the same size; $\Sigma_i = \Sigma$ constant across all clusters, where all clusters have the same geometry but need not be spherical; unrestricted $\Sigma_i$, where each cluster may have a different geometry. For $\Sigma_i = \lambda I$, only one parameter is required to characterize the covariance structure of the mixture, whereas $K(K+1)/2$ and $g(K(K+1)/2)$ parameters are needed for constant $\Sigma$ and unrestricted $\Sigma_i$.

To facilitate variable selection for high dimension, small sample size settings, which is often encountered in genomic studies, and to compare the standard model with the penalized model introduced later, we assume that all clusters have a common diagonal covariance matrix. Thus, the probability density function for cluster $i$ is
\[ f_i(x_j; \theta_j) = \frac{1}{(2\pi)^{K/2} |V|^{1/2}} \exp \left( -\frac{1}{2} (x_j - \mu_i)'V^{-1}(x_j - \mu_i) \right), \]

where \( V = \text{diag}(\sigma_1^2, \sigma_2^2, ..., \sigma_K^2) \), and \(|V| = \prod_{k=1}^{K} \sigma_k^2 \).

In the standard model-based clustering, first, the above mixture model is fitted to the data and thus the maximum likelihood estimate \( \hat{\Theta} \) is obtained. Second, the conditional probability of every observation \( x_j \) belonging to each of the \( g \) clusters can be calculated. Finally, each data point is assigned to the cluster with the largest posterior probability.

### 2.1.2 Estimation via EM algorithm

The fitting procedure is shown as follows. Suppose data \( x_j = \{x_{j1}, ..., x_{jk}\}, j = 1, ..., n \) are given, we want to maximize the log-likelihood

\[
\log L(\Theta) = \log \prod_{j=1}^{n} f(x_j; \Theta) = \sum_{j=1}^{n} \log \left[ \sum_{i=1}^{g} \pi_i f_i(x_j; \Theta) \right]
\]

with respect to \( \Theta \). Due to the numerical difficulty coming from the sum inside log, the maximum likelihood estimate has no closed form. It is common to utilize the expectation-maximization (EM) algorithm (Dempster et al., 1977) by casting the problem in the framework of missing data. In the EM for mixtures of Gaussian models, the complete data are considered to be \( y_j = (x_j, z_j) \), where \( z_j = (z_{j1}, z_{j2}, ..., z_{jg}) \) is the unobserved portion of the data, with

\[
z_{ij} = \begin{cases} 
1 & \text{if } x_j \text{ is from component } i \\
0 & \text{otherwise}
\end{cases}
\]

Assume that \( z_j \)'s are \textit{i.i.d.}, following a multinomial distribution of one drawn from \( g \) categories with parameters \( \pi_1, \pi_2, ..., \pi_g \). The density of an observation \( x_j \) given \( z_j \) is then given by

\[
f(x_j | z_j) = \prod_{i=1}^{g} f_i(x_j; \theta_i)^{z_{ij}}.
\]

The density of the complete data \( y_j \) is

\[
f(y_j) = f(x_j, z_j) = f(x_j | z_j) f(z_j) = \prod_{i=1}^{g} f_i(x_j; \theta_i)^{z_{ij}} \prod_{i=1}^{g} \pi_i^{z_{ij}} = \prod_{i=1}^{g} (\pi_i f_i(x_j; \theta_i))^{z_{ij}}.
\]
The resulting log-likelihood for the complete data can be represented as

\[
\log L_c(\Theta) = \log \prod_j f(y_j) = \log \prod_j \prod_i \left[ \pi_i f_i(x_j; \theta_i) \right]^{z_{ij}}
\]

\[
= \sum_i \sum_j z_{ij} \left[ \log \pi_i + \log f_i(x_j; \theta_i) \right]
\]

\[
= \sum_i \sum_j z_{ij} \left[ \log \pi_i - \frac{K}{2} \log 2\pi - \frac{1}{2} \sum_k \log \sigma_k^2 - \frac{1}{2} \sum_k \frac{(x_{jk} - \mu_k)^2}{\sigma_k^2} \right].
\]

The EM algorithm seeks to find the maximum likelihood estimate \( \hat{\Theta} = \{ \hat{\pi}_i, \hat{\mu}_k, \hat{\sigma}_k^2 : i = 1, ..., g; k = 1, ..., K \} \) by iteratively applying the following two steps:

**Expectation step**: compute the expected value of the above complete log-likelihood function, with respect to the conditional distribution of \( z \) given the data \( X \) and the current estimate of the parameters \( \hat{\Theta}^{(r)} \) (the estimate of \( \Theta \) at the \( r \)th iteration):

\[
Q(\Theta | \hat{\Theta}^{(r)}) = E_{z|X,\hat{\Theta}^{(r)}}(\log L(\Theta))
\]

\[
= \sum_i \sum_j \left[ \log \pi_i + \log f_i(x_j; \theta_i) \right] \hat{P}(z_{ij} = 1 | x_j) + \sum_i \sum_j 0 \left[ \log \pi_i + \log f_i(x_j; \theta_i) \right] \hat{P}(z_{ij} = 0 | x_j)
\]

\[
= \sum_i \sum_j \hat{\tau}_{ij}^{(r)} \left[ \log \pi_i + \log f_i(x_j; \theta_i) \right]
\]

where \( \hat{\tau}_{ij}^{(r)} \) is the posterior probability that \( x_j \) comes from \( i \)th component computed at the \( r \)th iteration:

\[
\hat{\tau}_{ij}^{(r)} = \hat{P}(z_{ij} = 1 | x_j) = \frac{\hat{P}(z_{ij} = 1, x_j)}{\hat{P}(x_j)} = \frac{\hat{P}(z_{ij} = 1) \hat{P}(x_j | z_{ij} = 1)}{\sum_{i=1}^g \hat{P}(z_{ij} = 1) \hat{P}(x_j | z_{ij} = 1)} = \frac{\hat{\pi}_i^{(r)} f_i(x_j; \hat{\theta}_i^{(r)})}{\sum_{i=1}^g \hat{\pi}_i^{(r)} f_i(x_j; \hat{\theta}_i^{(r)})}
\]

**Maximization step**: maximize the above quantity in terms of \( \pi_i \) and \( \theta_i \) with \( \hat{\tau}_{ij}^{(r)} \) being fixed at the values computed in the E-step

\[
\hat{\Theta}^{(r+1)} = \arg \max_{\Theta} Q(\Theta | \hat{\Theta}^{(r)}).
\]

Specifically,

\[
Q(\Theta | \hat{\Theta}^{(r)}) = \sum_i \sum_j \hat{\tau}_{ij}^{(r)} \left[ \log \pi_i + \log f_i(x_j; \theta_i) \right]
\]

\[
= \sum_i \sum_j \hat{\tau}_{ij}^{(r)} \left[ \log \pi_i - \frac{K}{2} \log 2\pi - \frac{1}{2} \sum_k \log \sigma_k^2 - \frac{1}{2} \sum_k \frac{(x_{jk} - \mu_k)^2}{\sigma_k^2} \right].
\]
We have

\[ \frac{\partial O}{\partial \pi_i} = \sum_j \frac{\hat{\tau}_{ij}}{\pi_i} - \sum_j \frac{\hat{\tau}_{ij}}{\pi_g} \quad i = 1, 2, ..., g - 1. \]

Set \( \frac{\partial O}{\partial \pi_i} = 0 \), then

\[ \hat{\pi}_i^{(r+1)} = \frac{\sum_j \hat{\tau}_{ij}}{\sum_j / \hat{\pi}_g^{(r+1)}} \quad i = 1, 2, ..., g - 1. \]

Since \( \sum_{i=1}^g \hat{\pi}_i = 1 \), therefore

\[ \hat{\pi}_i^{(r+1)} = \frac{\sum_{j=1}^n \hat{\tau}_{ij}}{n} \quad i = 1, 2, ..., g. \]

Calculate \( \hat{\mu}_{ik} \) as follows,

\[ \frac{\partial O}{\partial \mu_{ik}} = \frac{\sum_j \hat{\tau}_{ij} x_{jk} - \sum_j \hat{\tau}_{ij} \mu_{ik}}{\sigma_k^2} \quad i = 1, ..., g; \ k = 1, ..., K. \]

Set \( \frac{\partial O}{\partial \mu_{ik}} = 0 \), thus,

\[ \hat{\mu}_{ik}^{(r+1)} = \frac{\sum_{j=1}^n \hat{\tau}_{ij} x_{jk}}{\sum_{j=1}^n \hat{\tau}_{ij} }. \]

Similarly,

\[ \frac{\partial O_p}{\partial \sigma_k^2} = \sum_i \sum_j \tau_{ij}^{(r)} \left( -\frac{1}{2\sigma_k^2} + \frac{(x_{jk} - \hat{\mu}_{ik}^{(r)})^2}{2\sigma_k^4} \right) \quad k = 1, 2, ..., K. \]

Set \( \frac{\partial O_p}{\partial \sigma_k^2} = 0 \), thus

\[ \sigma_k^{2,(r+1)} = \sum_{i=1}^g \sum_{j=1}^n \hat{\tau}_{ij}^{(r)} (x_{jk} - \hat{\mu}_{ik}^{(r)})^2 / n. \]
We summarize the EM algorithm for the standard model-based clustering in the following:

**Step 1:** initialize parameters $\Theta^{(0)} = \{\pi^{(0)}_i, \mu^{(0)}_k, \sigma^{2,(0)}_k : i = 1, \ldots, g; k = 1, \ldots, K\}$ using the parameter estimates obtained from the $K$-means clustering;

**Step 2:** E-step calculates

$$
\hat{x}^{(r)}_{ij} = \frac{\sum_{i=1}^g \hat{\pi}_i^{(r)} f_i(x_j; \hat{\theta}_i^{(r)})}{\sum_{i=1}^g \sum_{j=1}^n \hat{\pi}_i^{(r)} f_i(x_j; \hat{\theta}_i^{(r)})};
$$

**Step 3:** M-step updates

$$
\hat{\pi}_i^{(r+1)} = \frac{\sum_{j=1}^n \hat{x}^{(r)}_{ij}}{n},
$$

$$
\hat{\mu}_k^{(r+1)} = \frac{\sum_{j=1}^n \hat{x}^{(r)}_{ij} x_{jk}}{\sum_{j=1}^n \hat{x}^{(r)}_{ij}},
$$

$$
\hat{\sigma}^{2,(r+1)}_k = \frac{\sum_{i=1}^g \sum_{j=1}^n (x_{jk} - \hat{\mu}_k^{(r)})^2}{n}.
$$

Repeat the above steps until convergence.

Under fairly mild regularity conditions, EM can be shown to converge to a local maximum of the observed data likelihood (Dempster *et al.*, 1977). Although these conditions do not always hold in practice, the EM algorithm has been widely used for maximum likelihood estimation for mixture models with good results.

### 2.2 Penalized model-based clustering

Clustering techniques have been widely applied in analyzing gene expression data. One of the most important issues is how to select informative genes out of thousands of those for the purpose of clustering, which is of more importance in a small sample size setting. The penalized model-based clustering was proposed to solve this problem.

#### 2.2.1 Penalizing mean parameters with an $L_1$ penalty

With the same motivation as in penalized regression for the purpose of variable selection, Pan and Shen (2007) proposed a penalized model-based clustering approach with an $L_1$ penalty. They showed that, with an appropriate choice of penalty function, it realized automatic selection and parameter estimation simultaneously.
Specifically, they regularize \( \log L(\Theta) \) by adding a penalty function to it, yielding a penalized log-likelihood function for the original (incomplete) data

\[
\log L_p(\Theta) = \sum_{j=1}^{n} \log \left( \sum_{i=1}^{g} \pi_i f_i(x_j; \theta_i) \right) - p_\lambda(\Theta) ,
\]

where \( L_p(\Theta) \) is the penalized likelihood function for the original data and \( p_\lambda(\Theta) \) is a penalty function with penalization parameter \( \lambda \). The form of \( p_\lambda(\Theta) \) depends on the goal of the analysis. The \( L_1 \) penalty is employed for variable selection as in the Lasso (Tibshirani, 1996). The corresponding penalized log-likelihood function for the complete data is:

\[
\log L_{c,p}(\Theta) = \sum_{i} \sum_{j} \log \left( \pi_i + \log f_i(x_j; \theta_i) \right) - p_\lambda(\Theta) .
\]

They proposed using such penalized model-based clustering as a general way to regularize parameter estimates, which can be useful for high-dimensional data, particularly those with relatively small sample sizes.

Pan and Shen (2007) proposed using a common diagonal covariance structure \( V \) across all clusters and the \( L_1 \) penalty \( p_\lambda(\Theta) = \lambda \sum_{i=1}^{K} | \mu_{ik} | \), where \( \mu_{ik} \) is the \( k \)th component of \( \mu_i \), the mean vector of component \( i \). We assume throughout this paper that we have standardized the data before doing clustering analysis, so that each attribute has sample mean 0 and sample variance 1. The main goal of using the \( L_1 \) penalty is to achieve a sparse solution with many small estimates of \( \mu_{ik} \) coerced to be 0, thus realizing variable selection automatically. For example, if \( \mu_{1k} = \mu_{2k} = \cdots = \mu_{gk} = 0 \), then the \( k \)th variable is non-informative in terms of clustering, since we also assume all clusters have a common diagonal covariance matrix. In this case, the \( k \)th variable will be treated as a noise variable and contribute nothing to the clustering analysis.

Next the EM algorithm for the above penalized model-based clustering is derived below. As mentioned before, \( \Theta = \{ \pi_1, \ldots, \pi_g, \mu_{i1}, \ldots, \mu_{ig}, \sigma_{1}^2, \ldots, \sigma_{K}^2 \} \) and \( \hat{\Theta}^{(r)} \) is used to represent the parameter estimates at iteration \( r \). Specifically,

The E-step yields:
\[ Q_p(\Theta | \hat{\Theta}^{(r)}) = E_{z|x, \hat{\Theta}^{(r)}}(\log \nu, p \ L(\Theta)) = \sum_i \sum_j \hat{\tau}_{ij}^{(r)} \left[ \log \pi_i + \log f_i(x_j; \theta_i) \right] - \lambda \sum_i \sum_k |\mu_{ik}| \]

\[ = \sum_i \sum_j \hat{\tau}_{ij}^{(r)} \left[ \log \pi_i - \frac{K}{2} \log 2\pi - \frac{1}{2} \sum_k \log \sigma_k^2 - \frac{1}{2} \sum_k \left( \frac{x_{jk} - \mu_{ik}}{\sigma_k^2} \right)^2 \right] - \lambda \sum_i \sum_k |\mu_{ik}|. \]

It is easy to check that adding a penalty does not change the estimate forms of \( \hat{\tau}_{ij} \):

\[ \hat{\tau}_{ij}^{(r)} = \hat{P}(z_{ij} = 1 | x_j) = \frac{\hat{\pi}_i^{(r)} f_i(x_j; \hat{\theta}_i^{(r)})}{\sum_{i=1}^g \hat{\pi}_i^{(r)} f_i(x_j; \hat{\theta}_i^{(r)})}. \]

The M-step maximizes \( Q_p(\Theta | \hat{\Theta}^{(r)}) \) to update the parameter estimates. It is easy to verify that adding a penalty involving \( \mu_{ik} \) only does not change the estimate forms of \( \hat{\tau}, \hat{\sigma}_k^2 \). Below is how to estimate \( \mu_{ik} \). The partial derivative of \( Q_p \) with respect to \( \mu_{ik} \) is

\[ \frac{\partial Q_p}{\partial \mu_{ik}} = \sum_j \hat{\tau}_{ij}^{(r)} \left( \frac{x_{jk} - \mu_{ik}}{\sigma_k^{2(r)}} \right) - \lambda \text{sign}(\mu_{ik}) \quad \text{if} \quad \mu_{ik} \neq 0. \]

Set \( \frac{\partial Q_p}{\partial \mu_{ik}} = 0 \), then

\[ \hat{\mu}_{ik}^{(r+1)} = \frac{\sum_j \hat{\tau}_{ij}^{(r)} x_{jk} - \lambda \hat{\sigma}_k^{2(r)} \text{sign}(\hat{\mu}_{ik}^{(r+1)})}{\sum_j \hat{\tau}_{ij}^{(r)}} \quad \text{if} \quad \mu_{ik} \neq 0. \]

Some calculations yield

\[ \frac{\sum_{j=1}^n \hat{\tau}_{ij}^{(r)} x_{jk}}{\sum_{j=1}^n \hat{\tau}_{ij}^{(r)}} = (| \hat{\mu}_{ik}^{(r+1)} | + \lambda \hat{\sigma}_k^{2(r)} \text{sign}(\hat{\mu}_{ik}^{(r+1)})). \]

Thus

\[ \text{sign} \left( \sum_{j=1}^n \hat{\tau}_{ij}^{(r)} x_{jk} \right) = \text{sign} \left( \hat{\mu}_{ik}^{(r+1)} \right). \]

Therefore, we can rewrite \( \hat{\mu}_{ik}^{(r+1)} \) as
\[
\hat{\mu}_{ik}^{(r+1)} = \frac{\sum_j \hat{z}_{ij}^{(r)} x_{jk} - \lambda_i \hat{\sigma}_k^{2(r)} \text{sign}(\hat{\mu}_{ik}^{(r+1)})}{\sum_j \hat{z}_{ij}^{(r)}}
\]
\[
= \frac{\sum_{j=1}^n \hat{z}_{ij}^{(r)} x_{jk}}{\sum_{j=1}^n \hat{z}_{ij}^{(r)}} \left(1 - \frac{\lambda_i \hat{\sigma}_k^{2(r)} \text{sign}(\hat{\mu}_{ik}^{(r+1)})}{\sum_{j=1}^n \hat{z}_{ij}^{(r)} x_{jk}}\right)
\]
\[
= \frac{\sum_{j=1}^n \hat{z}_{ij}^{(r)} x_{jk}}{\sum_{j=1}^n \hat{z}_{ij}^{(r)}} \left(1 - \frac{\lambda_i \hat{\sigma}_k^{2(r)}}{\left|\sum_{j=1}^n \hat{z}_{ij}^{(r)} x_{jk}\right|}\right) \quad \text{if } \mu_{ik} \neq 0 \quad (1).
\]

If \( \mu_{ik} = 0 \), then we compare \( Q_p(0, \cdot) \) and \( Q_p(\Delta \mu_{ik}, \cdot) \) when \( \Delta \mu_{ik} \) is near 0. By setting \( Q_p(0, \cdot) \geq Q_p(\Delta \mu_{ik}, \cdot) \), as \( \Delta \mu_{ik} \to 0 \), we can obtain,
\[
\left|\frac{\sum_{j=1}^n \hat{z}_{ij}^{(r)} x_{jk}}{\hat{\sigma}_k^{2(r)}}\right| \leq \lambda_i \quad (2).
\]

From (1) and (2), we have
\[
\hat{\mu}_{ik}^{(r+1)} = \frac{\sum_{j=1}^n \hat{z}_{ij}^{(r)} x_{jk}}{\sum_{j=1}^n \hat{z}_{ij}^{(r)}} \left(1 - \frac{\lambda_i \hat{\sigma}_k^{2(r)}}{\left|\sum_{j=1}^n \hat{z}_{ij}^{(r)} x_{jk}\right|}\right),
\]

where \( f_+ = \begin{cases} f & f > 0 \\ 0 & \text{otherwise} \end{cases} \).

Let \( \tilde{\mu}_{ik}^{(r+1)} = \sum_{j=1}^n \hat{z}_{ij}^{(r)} x_{jk} / \sum_{j=1}^n \hat{z}_{ij}^{(r)} \), where \( \tilde{\mu}_{ik}^{(r+1)} \) is the usual update for \( \mu_{ik} \) if we impose no penalty, therefore
\[
\hat{\mu}_{ik}^{(r+1)} = \tilde{\mu}_{ik}^{(r+1)} \left(1 - \frac{\lambda_i \hat{\sigma}_k^{2(r)}}{\left|\sum_{j=1}^n \hat{z}_{ij}^{(r)} x_{jk}\right|}\right)_+.
\]
It is evident from the updating formula of $\hat{\mu}_{ik}$ that, if $\lambda > \sum_{j=1}^{n} \hat{z}_{ij}^{(r)} x_{jk} / \hat{\sigma}_{k}^{2(r)}$, then $\hat{\mu}_{ik}^{(r+1)} = 0$; otherwise, $\hat{\mu}_{ik}^{(r+1)}$ is obtained by shrinking the usual estimate $\hat{\mu}_{ik}^{(r+1)}$ towards 0 by a percentage of $\lambda \hat{\sigma}_{k}^{2(r)} / \left| \sum_{j=1}^{n} \hat{z}_{ij}^{(r)} x_{jk} \right|$.

In summary, the EM algorithm for the penalized clustering with an $L_{1}$ penalty involves:

**Step 1:** initialize parameters $\Theta^{(0)} = \{\pi_{i}^{(0)}, \mu_{ik}^{(0)}, \sigma_{k}^{2,(0)} : i = 1, \ldots, g; k = 1, \ldots, K\}$ using the parameter estimates obtained from the $K$-means clustering;

**Step 2:** E-step calculates

$$\hat{z}_{ij}^{(r)} = \frac{\hat{p}_{i}^{(r)} f_{i}(x_{j} ; \hat{\theta}_{i}^{(r)})}{\sum_{i=1}^{g} \hat{p}_{i}^{(r)} f_{i}(x_{j} ; \hat{\theta}_{i}^{(r)})} ;$$

**Step 3:** M-step updates

$$\hat{\pi}_{i}^{(r+1)} = \frac{\sum_{j=1}^{n} \hat{z}_{ij}^{(r)} / n}{n},$$

$$\hat{\mu}_{ik}^{(r+1)} = \frac{\hat{\mu}_{ik}}{\hat{\mu}_{ik}} \left(1 - \frac{\lambda \hat{\sigma}_{k}^{2(r)}}{\sum_{j=1}^{n} \hat{z}_{ij}^{(r)} x_{jk}} \right),$$

$$\hat{\sigma}_{k}^{2,(r+1)} = \sum_{i=1}^{g} \sum_{j=1}^{n} \hat{z}_{ij}^{(r)} (x_{jk} - \hat{\mu}_{ik}^{(r)})^2 / n.$$

Repeat the above steps until convergence.

**2.2.2 Penalizing mean parameters with an $L_{p}$ penalty**

In this subsection, we extend the $L_{1}$ penalty to a $L_{p}$ penalty with $0 < p < 1$, where the $L_{p}$ penalty is defined as $L_{p} = \lambda_{p} \sum_{i} \sum_{k} | \mu_{ik} |^{p}$. The goal is to explore the shrinkage and variable selection of the generalized method and to check if there exists an optimal $p$, where the generalized one achieves the smallest modified BIC values defined in the next section.
Lemma 1: Suppose \( f(x) = ax^{2-p} - bx^1 - p + c \), where \( a \neq 0 \), \( b \), \( c \) are any real numbers, and \( 0 < p < 1 \), then there are at most two roots of \( f(x) = 0 \).

Proof: \( f'(x) = a(2-p)x^{1-p} - b(1-p)x^{-p} \). Set \( f'(x) = 0 \), then \( x = b(1-p) / a(2-p) \). Therefore, there are at most one critical number, resulting at most two roots of \( f(x) = 0 \).

Lemma 2: Suppose \( a = \text{sign}(x)a_0, \ b = \text{sign}(x)c_0 \), where \( a_0 > 0 \), \( c_0 > 0 \), and \( b \) is any number. Define \( f(x) = a \cdot |x^{2-p} - b| \cdot x^1 - p + c \), and \( f(x) = F'(x) \). Then,

1. if \( b > 0 \), and if there are two roots of \( f(x) = 0 \) on \((0, \infty)\), then there are two local maxima of \( F(x) \), one at \( x = 0 \), the other at the larger root of \( f(x) \); the one maximizing \( F(x) \) is the location of the absolute maximum;

2. if \( b < 0 \), and if there are two roots of \( f(x) = 0 \) on \((-\infty, 0)\), then there are two local maxima of \( F(x) \), one at \( x = 0 \), the other at the smaller root of \( f(x) \); the one maximizing \( F(x) \) is the location of the absolute maximum;

3. \( x = 0 \) is the unique maximizer of \( F(x) \) in all the other cases.

Proof: First, consider \( b > 0 \),

i) When \( x < 0 \), then \( a < 0 \) and \( c < 0 \), thus \( f(x) = a \cdot |x^{2-p} - b| \cdot x^1 - p + c < 0 \);

ii) When \( x > 0 \), then \( a > 0 \) and \( c > 0 \).

a) If there is no root of \( f(x) = 0 \) on \((0, \infty)\), since \( a > 0 \), then \( f(x) > 0 \). In this case, \( x = 0 \) is the unique maximizer of \( F(x) \);

b) If there is only one root of \( f(x) = 0 \) on \((0, \infty)\), since \( a > 0 \), then \( f(x) \geq 0 \). In this case, \( x = 0 \) is the unique maximizer of \( F(x) \);

c) If there are two roots of \( f(x) = 0 \) on \((0, \infty)\), then there are two local maxima of \( f(x) \), one at \( x = 0 \), the other at the larger root of \( f(x) = 0 \). The one maximizing \( F(x) \) is the location of the absolute maximum.

Second, consider \( b < 0 \),

i) When \( x > 0 \), then \( a > 0 \) and \( a > 0 \), thus \( f(x) = a \cdot |x^{2-p} - b| \cdot x^1 - p + c > 0 \);

ii) When \( x < 0 \), then \( a < 0 \) and \( c < 0 \).
a) If there is no root of \( f(x) = 0 \) on \((-\infty, 0)\), then \( f(x) < 0 \). In this case, \( x = 0 \) is the unique maximizer of \( F(x) \);

b) If there is only one root of \( f(x) = 0 \) on \((-\infty, 0)\), then \( f(x) \leq 0 \). \( x = 0 \) is the unique maximizer of \( F(x) \);

c) If there are two roots of \( f(x) = 0 \) on \((-\infty, 0)\), then there are two local maxima of \( F(x) \), one at \( x = 0 \), the other at the smaller root of \( f(x) = 0 \). The one maximizing \( F(x) \) is the location of the absolute maximum.

Finally, when \( b = 0 \), \( f(x) = a|x|^{p-1} + c \), \( f(x) > 0 \) if \( x > 0 \), and \( f(x) < 0 \) if \( x < 0 \). Therefore, \( x = 0 \) is the unique maximizer of \( F(x) \). QED

With the \( L_p \) penalty, the E-step yields

\[
Q_p = \sum_i \sum_j \tau_{ij}^{(r)} \left[ \log \pi_i + \log f_i(x_j; \theta_i) \right] - \lambda_i \sum_k |\mu_{ik}|^p
\]

\[
= \sum_i \sum_j \tau_{ij}^{(r)} \left[ \log \pi_i - \frac{K}{2} \log 2\pi - \frac{1}{2} \sum_k \log \sigma_k^2 - \frac{1}{2} \sum_k \frac{(x_{jk} - \mu_{ik})^2}{\sigma_k^2} \right] - \lambda_i \sum_k |\mu_{ik}|^p
\]

All the estimates except \( \hat{\mu}_{ik} \) obtained in the M-step are the same as those resulting from an \( L_1 \) penalty. As for \( \hat{\mu}_{ik} \), we derive the following theorem:

**Theorem 1:**

i) A necessary condition for \( \hat{\mu}_{ik} \) to be a local maximizer of \( Q_p \) is:

\[
\sum_j \tau_{ij}^{(r)} (x_{jk} - \hat{\mu}_{ik}) = \frac{\hat{\mu}_{ik}}{\sigma_k^2} = \lambda_i p |\mu_{ik}|^{p-1} \text{sign}(\mu_{ik}) \quad \text{if} \quad \hat{\mu}_{ik} \neq 0;
\]

ii) \( \hat{\mu}_{ik} = 0 \) is always a local maximizer of \( Q_p \).

**Proof:**

Proof of i):

If \( \hat{\mu}_{ik} \neq 0 \), then:

\[
\frac{\partial Q_p}{\partial \mu_{ik}} = \sum_j \tau_{ij}^{(r)} \frac{(x_{jk} - \mu_{ik})}{\sigma_k^2} - \lambda_i p |\mu_{ik}|^{p-1} \text{sign}(\mu_{ik})
\]
Proof of \( ii \) :

\[
\left( -\frac{\partial Q_p}{\partial \mu_{ik}} \right) | \mu_{ik} |^{1-p} = -\sum_j \tau_{ij}^{(r)} \left( \frac{x_{jk} - \mu_{ik}}{\hat{\sigma}_k^2} \right) | \mu_{ik} |^{1-p} + \lambda_i \text{sign}(\mu_{ik}) \\
= \frac{1}{\hat{\sigma}_k^2} \text{sign}(\mu_{ik}) | \mu_{ik} |^{2-p} - \frac{\sum_j \tau_{ij}^{(r)} x_{jk}}{\hat{\sigma}_k^2} | \mu_{ik} |^{1-p} + \lambda_i \text{sign}(\mu_{ik}).
\]

Let \( a_{ik} = \text{sign}(\mu_{ik}) \sum_j \tau_{ij}^{(r)} / \hat{\sigma}_k^2 \), \( b_{ik} = \sum_j \tau_{ij}^{(r)} x_{jk} / \hat{\sigma}_k^2 \), \( c_{ik} = \lambda_i \text{sign}(\mu_{ik}) \), and define

\[
f(\mu_{ik}) = a_{ik} | \mu_{ik} |^{2-p} - b_{ik} | \mu_{ik} |^{1-p} + c_{ik}.
\]

Then we apply lemma 2 and obtain that \( \hat{\mu}_{ik} = 0 \) is always a local maximizer of \( Q_p \). \( \text{QED} \)

We do not have a simple formula to update \( \hat{\mu}_{ik} \). Using the same notation defined in the proof of theorem 1, we apply lemma 2 and obtain the solution of \( \hat{\mu}_{ik} \). Specifically,

1. if \( b_{ik} > 0 \), and there are two roots of \( f(\mu_{ik}) = 0 \) on \( (0, \infty) \), then there are two local maxima of \( Q_p \), one at \( \hat{\mu}_{ik} = 0 \), the other at the larger root of \( f(\mu_{ik}) = 0 \); the one maximizing \( Q_p \) is chosen to update \( \hat{\mu}_{ik} \);

2. if \( b_{ik} < 0 \), and there are two roots of \( f(\mu_{ik}) = 0 \) on \( (-\infty, 0) \), then there are two local maxima of \( Q_p \), one at \( \hat{\mu}_{ik} = 0 \), the other at the smaller root of \( f(\mu_{ik}) = 0 \); the one maximizing \( Q_p \) is chosen update \( \hat{\mu}_{ik} \);

3. \( \hat{\mu}_{ik} = 0 \) is the unique maximizer of \( Q_p \) in all the other cases.

Theoretically, we can use the above formulas to update \( \hat{\mu}_{ik} \). However, it is obvious from the theorem that \( \hat{\mu}_{ik} \) would be zero in most cases. If \( \hat{\mu}_{ik} \) not equal to zero, then \( \hat{\mu}_{ik} \) has to be a root of \( f(x) \), which could only be obtained by numerical approximation methods. Hence, there is no clear relationship between nonzero \( \hat{\mu}_{ik} \) and \( \hat{\mu}_{ik} \), the usual estimate of \( \mu_{ik} \) in the standard model-based clustering. That is, there is no clear shrinkage if \( \hat{\mu}_{ik} \neq 0 \). Therefore, we conjecture that adding a penalty with an \( L_p \) \( (0 < p < 1) \) norm form to the standard mixture model may not perform as well as the penalized model with an \( L_1 \) penalty, where there is clearly a shrinkage, although no numerical examples are provided in this paper to justify our thoughts.
2.3 Model selection

An important issue in applied cluster analysis is the determination of the number of clusters. In a range of applications of model-based clustering, model choice based on the Bayesian information criterion (BIC) (Schwarz, 1978) has given good results (Fraley and Raftery 1998). BIC is defined as

\[ BIC = -2 \log L(\hat{\Theta}) + \log(n)d \]

where \( \hat{\Theta} \) is the maximum likelihood estimate, \( d = \text{dim}(\Theta) \) is the total number of unknown parameters. In the standard model-based clustering with a common covariance matrix setting, \( \Theta = \{\pi_1, \ldots, \pi_g, \mu_{11}, \ldots, \mu_{gK}, \sigma_{11}^2, \ldots, \sigma_K^2\} \) with one restriction \( \sum_{i=1}^{g} \pi_i = 1 \). Hence, \( d = g + K + gK - 1 \).

Intuitively, the first term in the definition of BIC, which is the maximized mixture likelihood for the model, rewards a model that fits the data well, and the second term discourages overfitting by penalizing models with more free parameters. A small BIC score indicates strong evidence for the corresponding model.

For the penalized model-based clustering, following a conjecture of Efron et al. (2004) and the result of Zou et al. (2004) for \( L_1 \) penalized regression, Pan and Shen (2007) proposed a modified BIC defined as

\[ BIC = -2 \log L(\hat{\Theta}) + \log(n)d_e \]

where \( d_e = g + K + gK - 1 - q \) is the effective number of parameters. They set \( q \) as the number of the mean components that equal to 0. That is, \( q = \#\{k : \mu_{1k} = \mu_{2k} = \cdots = \mu_{gk} = 0\} \). Therefore, \( d_e \) is the number of non-zero parameter estimates.

In applying the penalized approach to the gene expression data, we fit a series of models with \( g = 1, 2, \ldots, 9 \). For each \( g \), we use various penalization parameters \( \lambda = 1, 2, 5, 7.5, 10, 15 \) and 20. For each combination of \( g \) and \( \lambda \), we apply the K-means first and use K-means’ results as an initialization for the EM. We select a \( g \) with the minimum BIC as the optimal \( g \).

2.4 Independent assessment of clusters

A clustering result can be considered as a partition of objects into groups. Hence, comparing a clustering result to the external criterion is equivalent to assessing the agreement of two partitions. The Rand index (Rand, 1971) is a measure of similarity between two clusterings. It is defined as the fraction of agreement, i.e. the number of pairs of objects that are either in the same groups in both partitions or in different groups in both partitions, divided by the total number of pairs of objects. The Rand index lies between 0 and 1, with 1 indicating that the two clusters are
exactly the same and 0 indicating that the two data clusters do not agree on any pair of points. The adjusted Rand index (Hubert and Arabie, 1985) is the corrected-for-chance version of the Rand index. A high adjusted Rand index indicates a high level of agreement between the two partitions.

Suppose we have a data set \( X = \{x_1, \ldots, x_n\} \) and two clusterings of \( X \), \( P = \{p_1, \ldots, p_r\} \) and \( Q = \{q_1, \ldots, q_s\} \), we define the following:

\( a \): the number of pairs of elements in \( X \) that are in the same set in \( P \) and in the same set in \( Q \);
\( b \): the number of pairs of elements in \( X \) that are in different sets in \( P \) and in different sets in \( Q \);
\( c \): the number of pairs of elements in \( X \) that are in the same set in \( P \) and in different sets in \( Q \);
\( d \): the number of pairs of elements in \( X \) that are in different sets in \( P \) and in the same set in \( Q \).

The Rand index (RI) is:

\[
RI = \frac{a + b}{a + b + c + d} = \frac{a + b}{\binom{n}{2}}.
\]

The adjusted Rand index (aRI) is:

\[
aRI = \frac{2(ab - cd)}{(a + d)(d + b) + (a + c)(c + b)}.
\]
Chapter 3

Examples

In this section, the model-based clustering and $L_1$ penalized approach are applied to two real gene expression data sets, and the results are shown. To compare the performances of these two methods, we assume each cluster shares a common diagonal covariance matrix. The MCLUST package in R is used to implement model-based clustering. Currently, no package is available for implementing the penalized approach. This paper uses R to realize the penalized clustering.

3.1 CNS tumors data

Five distinct types of embryonal tumors of the central nervous system (CNS) were studied by Pomeroy et al. (2002), including 10 medulloblastomas (MD); 8 primitive neuroectodermal tumors (PNET); 10 atypical teratoid/rhabdoid tumors (Rhab); 10 malignant gliomas (Glio); and 4 normal cerebellum (Ncer). There are 1000 genes for each tumor. Our objective is to show the potential usefulness of the two model-based clustering methods. Due to the default precision of R and the inefficiency of our coding, we only select the first 100 genes. We apply these two methods to the tumors data; each tumor is treated as an observation whiles genes are treated as attributes. For each method, $g$ goes from 1 to 9, and the $g$ with a minimum BIC is chosen.

![Figure 1. BIC scores and adjusted Rand indices for the clustering analysis of CNS tumors data](image)

The clustering results are shown in Figure 1. The penalized clustering performs better than the standard one. Figure 1 (left) shows the BIC values of both methods over a range of different numbers of clusters. Obviously, BIC scores of penalized model-based clustering are smaller than those of the standard one across all $g$’s. Both methods reach their minimum BIC values at
$g = 6$, which is not in line with the true number of clusters $g = 5$. However, both approaches achieve their maximum adjusted Rand indices at $g = 5$. The maximum adjusted Rand index of the penalized model-based clustering is 0.583, while that of the standard one is 0.502. Hence, it seems reasonable to think that these clusters are relatively close to each other, which makes it harder to separate them apart. Similar as the “elbow” used in determining the number of components in principle component analysis, there is an “elbow” at $g = 4$ in the BIC figure. We would choose $g = 4$ if we use “elbow” as the only criterion. The reason why there is such an “elbow” at $g = 4$ may be that the clusters of tumors data are relatively close and the sample size of the normal cerebellum cluster in tumors data is only 4 while the other 4 clusters have sample sizes around $n = 10$, so the normal cerebellum cluster may be distributed into the other clusters. In penalized clustering, about 27% of the mean parameter estimates are 0’s, but only 6 genes have their cluster-specific mean estimates as 0’s across all 5 clusters, and thus are treated as noise variables.

### 3.2 Novartis multi-tissue data

We select a subset of the Novartis multi-tissue samples (Su et al., 2002). The subset consists of four distinct cancer types: 6 breast, 6 prostate, 6 lung and 6 colon; 100 out of 1000 genes are used.

![Figure 2. BIC scores and adjusted Rand indices for the clustering analysis of Norvartis multi-tissue data](image)

Figure 2 (left) shows that both approaches achieve their minimum BIC at $g = 4$, which is in line with the true number of clusters. Moreover, the penalized clustering produces smaller BIC values across all clusters. In the adjusted Rand index figure, there is a peak for each clustering at
$g = 4$, and each one reaches the maximum adjusted Rand index value $1$. The Rand indices of both methods are the same at $g = 1, 2, 3$ and 4, but the Rand indices of the penalized clustering are a little bigger than those of the standard one when $g$ is greater than 4. Based on Figure 2, we conclude that the penalized model-based clustering performs slight better than the standard one in this case. In penalized clustering, about 29% of the mean parameter estimates are 0’s, but only 4 genes have their cluster-specific mean estimates as 0’s across all 4 clusters, and thus are treated as non-informative.
Chapter 4

Discussion

In this paper, we reviewed the standard model-based clustering and the $L_1$ penalized clustering, and showed theoretically there was no clear shrinkage property in the $L_p$ penalized approach. Xie et al. (2008) generalized the penalized clustering with an $L_1$ penalty involving mean parameters only to the one involving both mean and variance parameters in the context of mixtures of Gaussian with cluster-specific diagonal covariance matrices. Since our goal here is to investigate the shrinkage property of the $L_p$ penalized approach, we did not introduce their work.

Due to the inefficiency of our codes, we did not use all the information, and only selected 100 genes out of 1000 for both data. Even so, when $g$ was greater than 3, the computation went slow due to the iterations in the EM step. However, we still showed that the penalized model-based clustering produces slightly higher quality clusters than the standard approach, at least for our two real gene expression data sets. Pan and Shen (2007) applied both methods to a dataset with 2000 genes and 38 observations. They also obtained smaller BIC values for the penalized clustering, but the clusters produced by these two methods were of the similar quality.

When the sample size is moderately large, we can take a random sample of the data and apply these methods to the sample. The results are then extended to the full dataset using discriminant analysis, with the sample viewed as the training set.

In penalized methods, the choice of the penalization parameter is also important. We used various values of penalization parameter and utilized the modified BIC criterion to select a best one. For different datasets and different $g$, the choice of the penalization parameter is different. Although it seemed to work well in our examples, further evaluations are needed.
Acknowledgement

I would like to acknowledge with special thanks the help, both in this paper and in my personal life, from my advisor Dr. Kang Ling James, who brings me to a new level in understanding Statistics. I also would like to thank Dr. Barry James for his wonderful classes, and I am grateful for Dr. Yongcheng Qi and Dr. Steven Trogdon for their suggestions on my paper. I would like to thank Dr. Marshall Hampton for his reviews. Finally, I am grateful for the suggestions from my committee members: Dr. Kang Ling James, Dr. Barry James and Dr Marshall Hampton.
References


Appendix

1) Code for standard model-based clustering

library(mvtnorm)

#load package “mvtnorm”
# available at http://cran.r-project.org/web/packages/mvtnorm/index.html

library(mclust)

#load package “mclust”
# available at http://cran.r-project.org/web/packages/mclust/index.html

norva <- read.csv("F:\pnewnovaris.csv",header=FALSE, sep="","")

#read in standardized Novartis multi-tissue data

norva <- norva[c(1:6,27:32,53:58,81:86),1:100]

#select only 24 samples, with 6 samples for each cluster

norvaBIC.EEI <- mclustBIC(norva,modelNames = "EEI")

#standard model-based clustering analysis of data, with a common covariance structure for all clusters

plot(norvaBIC.EEI, xlab="number of components", ylab="BIC", type="b", col = "red", lwd=2,
panel.first =grid(8,8), pch=19,lty=1)

#plot BIC values

trueclass <- c(rep(0,6),rep(1,6),rep(2,6),rep(3,6))

#true classification

randindex.EEI <- numeric( )

for(g in 1:9) {
    randindex.EEI[g]=adjustedRandIndex(summary(norvaBIC.EEI,norva,
    G=g)$classification, truecluster)
}


# calculate adjusted Rand index for standard model-based clustering result

plot ( randindex.EEI,  xlab="number of components",  ylab="Adjusted Rand Index", type="b", col= "red", lwd=2, panel.first =grid(8,8), pch=19,lty=2)

#plot adjusted Rand index

2) **Code for L1 penalized model-based clustering with g=2 and lamda=1**

library(mvtnorm)       #load package “mvtnorm”

library(mclust)            #library(mclust)

norva <- read.csv("F:\pnewnovaris.csv",header=FALSE, sep="","")

kmeans.result <- kmeans(norva, centers=2)       # standardized Novartis multi-tissue data

# apply K-means with 2 centers

p1 <- sum(kmeans.result$cluster==1)/24
p2 <- sum(kmeans.result$cluster==2)/24

# setting up two priors, since we set g=2

theta <- list(  
    tau=c(p1,p2),                                       # tau is a vector of priors
    mu1=kmeans.result$centers[1,],         # mean parameters for cluster 1
    mu2=kmeans.result$centers[2,],         # mean parameters for cluster 2
    sig = diag(1,100)                                 # common covariance matrix
)

# theta is first set up as a vector of initial parameters using the results from K-means

lamda=1               # lamda is the penalization parameter

i=2                       # set # of clusters to be 2

k=100                  # set dimension to be 100

E.step <- function(theta)
t(apply(cbind(theta$tau[1] * dmvnorm(norva,mean=theta$mu1,sigma=theta$sig),
theta$tau[2] * dmvnorm(norva,mean=theta$mu2,sigma=theta$sig)),1,function(x) x/sum(x)))

# calculate conditional probabilities \hat{\tau}_{ij}

den <- numeric() # set denominator of \hat{\mu}_y \left( \sum_{j=1}^{n} \hat{\tau}_{ij}^{(r)} \right) to be numeric

for(i in 1:2) {
    den[i]=sum(E.step(theta)[,i])
}

num <- c(1:(i*k)) # set numerator of \hat{\mu}_y \left( \sum_{j=1}^{n} \hat{\tau}_{ij}^{(r)} \chi_{jk} \right) to be a matrix

attr(num,"dim") <- c(i,k)

for(i in 1:2) {
    for(k in 1:100) {
        num[i,k]=sum(E.step(theta)[,i]*norva[,k])
    }
}

sig.in <- diag(3,k,k)

for(j in 1:10000) {
    sig.in[j]=theta$sig[j]
}

brace <- c(1:(i*k)) # use brace to store the values of \left( 1 - \frac{\lambda_0 \hat{\sigma}_k^{2, (r)}}{\sum_{j=1}^{n} \hat{\tau}_{ij}^{(r)} \chi_{jk}} \right)

attr(brace,"dim") <- c(i,k)

for(i in 1:2) {


for(k in 1:100) {
    brace[i,k]=1-lamda*sig.in[k]/abs(num[i,k])
}

mu <- c(1:(i*k)) # use mu to store $\hat{\mu}_i$, so mu is set up as a matrix

attr(mu,"dim") <- c(i,k)

for(i in 1:2) {
    for(k in 1:100) {
        mu[i,k]=num[i,k]/den[i]*ifelse(brace[i,k]>0,brace[i,k],0)
    }
}

# Recall $\hat{\mu}_i^{(r+1)} = \frac{\sum_{j=1}^n \hat{r}_{ij}^{(r)} x_{jk}}{\sum_{j=1}^n \hat{r}_{ij}^{(r)}} \left(1 - \frac{\lambda_i \hat{\sigma}^2_{ik}^{(r)}}{\left|\sum_{j=1}^n \hat{r}_{ij}^{(r)} x_{jk}\right|}\right)$,

sigk <- numeric( ) # use sigk to store $\hat{\sigma}^2_{ik}$

for(k in 1:100) {
    sigk[k]=sum( 
        E.step(theta)[,1]*((norva[,k]-theta$mu1[k])^2) 
        +E.step(theta)[,2]*((norva[,k]-theta$mu2[k])^2) 
    )/nrow(norva)
}

M.step <- function(T) list( 
    tau= apply(T,2,mean),
    mu1= mu[1,],
    mu2= mu[2,],

    ...
```r
sig = diag(sigk, 100)

# M.step updates estimates
likelihood <-
  sum(log(
    theta$tau[1] * dmvnorm(norva, mean = theta$mu1, sigma = theta$sig) +
    theta$tau[2] * dmvnorm(norva, mean = theta$mu2, sigma = theta$sig)
  )
)

# define log-likelihood
q <- 0  # define # of noise variables
dimension <- 100 + 24 + 100 * 24 - 1 - q  # degree of freedom
BIC1 <- 2 * likelihood - log(nrow(norva)) * dimension  # set up initial BIC score
BIC2 <- 0  # set up intermediate BIC score

while (abs(BIC1 - BIC2) > 0.01) {
  # EM algorithm repeating E-step and M-step, with threshold to be 0.01
  BIC1 <- BIC2  # BIC1 is the old BIC
  T <- E.step(theta)  # posterior probabilities
den <- numeric()
  for(i in 1:2) {
    den[i] = sum(E.step(theta)[,i])
  }
  num <- c(1:(i*k))
```
```r
attr(num,"dim") <- c(i,k)
for(i in 1:2) {
    for(k in 1:100) {
        num[i,k]=sum(E.step(theta)[,i]*norva[,k])
    }
}
sig.in <- diag(3,k,k)
for(j in 1:10000) {
    sig.in[j]=theta$sig[j]
}
brace <- c(1:(i*k))
attr(brace,"dim") <- c(i,k)
for(i in 1:2) {
    for(k in 1:100) {
        brace[i,k]=1-lamda*sig.in[k]/abs(num[i,k])
    }
}
mu <- c(1:(i*k))
attr(mu,"dim") <- c(i,k)
for(i in 1:2) {
    for(k in 1:100) {
        mu[i,k]=num[i,k]/den[i]*ifelse(brace[i,k]>0,brace[i,k],0)
    }
}
```
sigk <- numeric()

for(k in 1:100) {
    sigk[k] = sum(E.step(theta)[,1]*((norva[,k]-theta$mu1[k])^2) + E.step(theta)[,2]*((norva[,k]-theta$mu2[k])^2) )/nrow(norva)
}

theta <- M.step(T)

likelihood <- sum(log(theta$tau[1]*dmvnorm(norva, mean=theta$mu1, sigma=theta$sig) + theta$tau[2]*dmvnorm(norva, mean=theta$mu2, sigma=theta$sig)))

q <- length(which(theta$mu1==theta$mu2))
dimension <- 100+24+100*24-1-q

BIC2 <- 2*likelihood-log(nrow(norva))*dimension

}  # end of EM algorithm

cluster <- numeric()  # cluster result

for(j in 1:24) {
    cluster[j] = which(E.step(theta)[j,]==max(E.step(theta)[j,]))
}

cluster
truecluster <- c(rep(0,6),rep(1,6),rep(2,6),rep(3,6))

randindex.pmbc <- adjustedRandIndex(cluster,truecluster)

randindex.pmbc  # adjusted Rand index for L1 penalized model

3) Code for plotting BIC values and adjusted Rand index for both models

norvaPBIC <- c(6940.133,5810.198,5614.436, 5235.967, 5339.708, 5463.475,5558.574,5564.387,5675.642)

# BIC values for L1 penalized clustering
plot(norvaPBIC, xlab="number of components", ylab="BIC", type="b", ylim=c(5200, 7100),col="red", lwd=2, panel.first =grid(8,8), pch=19,lty=1)
lines(c(1,2),c(6994.143,5977.560),col="blue", pch=24,type="b", lwd=2, lty=2)
# adding the points of standard clustering
lines(c(2,3),c(5977.560,5766.419),col="blue", pch=24,type="b", lwd=2, lty=2)
lines(c(3,4),c(5766.419,5508.646),col="blue", pch=24,type="b", lwd=2, lty=2)
lines(c(4,5),c(5508.646,5599.860),col="blue", pch=24,type="b", lwd=2, lty=2)
lines(c(5,6),c(5599.860,5644.613),col="blue", pch=24,type="b", lwd=2, lty=2)
lines(c(6,7),c(5644.613,5644.157),col="blue", pch=24,type="b", lwd=2, lty=2)
lines(c(7,8),c(5644.157,5675.490),col="blue", pch=24,type="b", lwd=2, lty=2)
lines(c(8,9),c(5675.490,5777.191),col="blue", pch=24,type="b", lwd=2, lty=2)
legend(4,6500,c("Penalized model based clustering","Standard model based clustering"),lty=c(1,2),lwd=2,col=c("RED","BLUE"))

PMBC.randindex <- c(0.0000000, 0.3030303, 0.6849315, 1.0000000, 0.9451074, 0.8986784, 0.8498242, 0.7850467, 0.6866485)

# Rand index values for L1 penalized clustering
plot(PMBC.randindex, xlab="number of components", ylab="Adjusted Rand Index",type="b", col= "red", lwd=2, panel.first =grid(8,8), pch=19,lty=1)
lines(c(1,2),c(0.0000000,0.3030303),col="blue", pch=24,type="b", lwd=2, lty=2)