Application of Gap statistics to Penalized
Model-Based Clustering

Wencheng Geng
University of Minnesota Duluth
Advisor: Prof. Kang James
May 21, 2010
Acknowledgements

First and foremost I want to thank my advisor Dr. Kang James. She helped me to locate my project topic and contributed so much time and ideas to help me in both my graduate study and research. Without her help, I could never have finished this project nor could I continue my career in the area of Statistical Analysis. I also very much appreciate the training of presentation skills from her.

Besides my advisor, I would like to express my sincere thanks to the rest of my committee. Dr. Barry James helped to review and correct my paper. His fantastic probability courses have always been an inspiration to me. I also want to thank Dr. Marshall Hampton for his review and suggestions.

The Department of Mathematics and Statistics at the University of Minnesota Duluth has provided me with excellent academic and hardware facilities throughout my graduate study.

Last but not least, I owe my warmest thanks to all my friends at UMD for their continuous support and help during these years.
Abstract

Clustering analysis is widely used in many areas such as data mining and genomic discoveries. For high dimensional data with the presence of many non-informative noise variables, elimination of those noise variables is crucial to better clustering of the data. When the number of clusters is known, the penalized model-based clustering realizes variable selection and parameter estimation simultaneously. However, determining an optimal number of clusters can be difficult and computationally demanding. We propose incorporating Gap statistics into clustering analysis by estimating the number of clusters before applying the penalized model-based method. We evaluate this approach with both simulation study and real data analysis.

Keywords: Clustering analysis, EM algorithm, Gap statistic, Model-based, Normal mixtures, Penalized likelihood, Variable selection.
# Table of Contents

1 Introduction ................................................. 1
2 Models ....................................................... 4
   2.1 Model-based clustering ........................................ 4
   2.2 Penalized model-based clustering ........................... 7
   2.3 Gap statistics .............................................. 9
   2.4 Model selection ............................................ 10
3 Simulations .................................................. 12
   3.1 Simulation set-ups .......................................... 12
   3.2 Results .................................................. 12
4 Examples ..................................................... 16
   4.1 Novartis multi-tissue data ................................. 16
   4.2 Lung cancer data .......................................... 18
5 Discussion .................................................... 21
References ..................................................... 22
Appendix ....................................................... 23
List of Tables

Table 1 Simulation results with set-up 1 ................................................. 13
Table 2 Simulation results with set-up 2 ................................................. 13
Table 3 Sample Novartis multi-tissue data ............................................. 17
Table 4 Sample lung cancer data ......................................................... 19
List of Figures

Figure 1 Modified BICs for $\lambda = 1, 5, 6, 7, 8, 9, 10, 15, 20$ with set-up 1 . . . . . . . . . . . . 13
Figure 2 Modified BICs for $\lambda = 1, 5, 6, 7, 8, 9, 10, 15, 20$ with set-up 2 . . . . . . . . . . . . 14
Figure 3 Modified BICs for $\lambda = 1, 2, 3, 4, 5$ with Novartis multi-tissue data . . . . . . . . 18
Figure 4 Modified BICs for $\lambda = 1, 2, 3, 4, 5$ with lung cancer data . . . . . . . . . . . . . . . . 19
1 Introduction

Determining the structure of clustered data is a fundamental task in many applications such as data mining, bioinformatics and genomic studies. For example, groups of functionally related genes may exhibit similar expression patterns and are therefore clustered together. A tremendous amount of effort has been made to categorize genes into groups and identify the groups involved in certain diseases or biological processes. In such applications, clustering analysis is an important and widely used method.

In clustering analysis, data are modeled by a mixture of components with each representing a different cluster. Observations in the same cluster are cohesive and separated from other clusters. Model-based clustering further assumes that the observations are drawn from a finite mixture of underlying probability distributions such as normal distributions. With this extra assumption, the work of determining a good clustering algorithm with appropriate number of clusters can be reduced to a model selection problem under the probability framework. It also allows the estimation of parameters of interest. Maximum likelihood is the common tool in parameter estimation, but direct estimation can be difficult. EM algorithm is usually applied to obtain the maximum likelihood estimators in such problems.

With the advent of high-throughput technologies, scientists have been able to record a great amount of data at the same time, while the analysis of such large datasets poses new challenges to statisticians. The emerging microarray technology enables biomedical researchers to monitor the expression levels of thousands of genes. However, microarray experiments are still expensive to perform, and only a few replicates are obtained in practice. This is the situation of “high dimension, low sample size” or the so-called “large p, small n”, where the data dimension or number of attributes greatly exceeds the number of observations. Among the many attributes, some only introduce noise and sometimes can even mask the cluster pattern. These attributes are called non-informative variables. Eliminating the non-informative variables improves both the quality of clustering analysis and model interpretability. Therefore, selecting informative variables becomes crucial to clustering analysis. In genomic studies, it is common to have a large proportion
of non-informative genes. Failure to include some important genes may impede the discrimination of phenotypes of interest, while noise introduced by the non-informative genes that are not excluded can sometimes lead to incorrect conclusions or even make the uncovering of true cluster structure difficult or impossible.

Unlike in other problems, many commonly used variable selection techniques such as forward and/or backward elimination or best subset selection can be unrealistic in clustering analysis [1]. Too many models have to be considered with high-dimensional data. Even if the dimension is fairly low, those techniques may select a model which is of no interest. Penalized likelihood approaches can be applied to handle these problems. Furthermore, with such approaches, variable selection and parameter estimation are realized simultaneously. In regression and classification, penalized methods have shown great success in variable selection [2]. In the light of this success, Pan and Shen [3] proposed a penalized model-based clustering method in the context of normal mixture models with a common diagonal covariance matrix across all clusters. In their approach, cluster means are adaptively shrunk towards the global mean. With an appropriate $L_1$ penalty function, means of the non-informative or noise components are estimated to be exactly equal to the global mean and therefore identified and excluded. Xie et al. [4] extended this method to the case with a cluster-specific diagonal covariance matrix, i.e., variances are different across clusters but same within each cluster.

In penalized model-based clustering, number of clusters in a dataset is unknown, and determining this number can be a very difficult task. Although prior knowledge of the number of components or other information about data composition can be obtained in many data mining areas, such knowledge is generally not available in microarray studies despite the rapid growth of modern biological sciences. To estimate the optimal number of components, Pan and Shen and Xie et al. used grid search with a modified Bayesian information criterion (BIC) as their model selection criterion. They first fit a series of models with different numbers of clusters, and then use the selected criterion to choose the best one. This approach is very computationally demanding and could limit the usage of their proposed method. In order to reduce the computational cost, we consider
estimating the number of clusters prior to variable selection through the penalized clustering.

Gap statistics were proposed by Hastie and Tibshirani et al. [5] to estimate the number of clusters in clustering analysis. It compares the change in within-cluster dispersion to that expected under a reference distribution. The optimal number of clusters should be chosen to maximize this change, or in other words, the “gap”. We propose to incorporate Gap statistics into penalized model-base clustering analysis to achieve better clustering with significantly reduced computational burden. Specifically, we break the problem into two parts: adopt Gap statistics to estimate an appropriate number of clusters, and then apply Pan and Shen’s work to realize selection of informative variables and estimation of parameters of interest. For simplicity, we assume a common diagonal covariance matrix.

The rest of the paper is arranged as follows. In Chapter 2, we first introduce model-based clustering as well as penalized model-based clustering. Maximum likelihood estimation with EM algorithm is used in fitting the model. The Gap statistic and its use in choosing the number of clusters are also discussed. We then evaluate our approach with simulated and real data in Chapters 3 and 4. A short discussion and future work are presented in Chapter 5.
2 Methods

Denote a random sample of $n$ $P$-dimensional observations by $X = \{x_i : j = 1, 2, ..., n\}$, where $x_i = \{x_{ip} : p = 1, 2, ..., P\}$, and $x_{ip}$ is the measurement of the $p$-th attribute. The goal of clustering is to partition the observed data with previously unknown structure into a set of exhaustive and non-overlapping clusters so that observations in the same cluster are more similar to each other than to those from other clusters.

2.1 Model-based clustering

In model-based clustering, each cluster is represented by a parametric distribution, such as normal distribution in Gaussian mixture models (GMM). It is assumed that each observation $x_i$ is independently drawn from a finite mixture distribution with probability density function

$$f(x_i; \Theta) = \sum_{k=1}^{K} \pi_k f_k(x_i; \theta_k),$$

where $\Theta$ represents all unknown parameters $\{(\pi_k, \theta_k) : k = 1, 2, ..., K\}$, and $\pi_k$'s are mixing proportions or prior probabilities of cluster $k$ with $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^{K} \pi_k = 1$. Under GMM, $f(x_i; \theta_k)$ is the multivariate normal density function corresponding to cluster $k$ with mean vector $\mu_k$ and covariance matrix $V_k$.

Data sampled from normal mixtures are characterized by clusters centered around $\mu_k$. The geometric features such as shape, volume and orientation of those clusters are determined by the structure of covariance matrix $V_k$. Meanwhile, the amount of unknown parameters also depends on $V_k$. For $V_k = \lambda I$ where all clusters are spherical and of the same size, only one parameter is required to characterize the covariance structure; for $V_k = V$ across all clusters with the same geometry but not necessarily spherical, $P(P+1)/2$ parameters are needed, whereas $KP(P+1)/2$ parameters are required for unrestricted $V_k$ where each cluster has a different geometry. To facilitate variable
selection with high dimensional data and for simplicity, we assume that all clusters have a common diagonal covariance matrix $V$. Thus, the probability density function for cluster $k$ is

$$f_k (x; \theta_k) = \frac{1}{(2\pi)^{p/2}|V|^{1/2}} \exp \left(-\frac{1}{2}(x - \mu_k)'V^{-1}(x - \mu_k)\right),$$

where $V = \text{diag}(\sigma_1^2, \sigma_2^2, ..., \sigma_p^2)$, and $|V| = \prod_{p=1}^p \sigma_p^2$.

To obtain the maximum likelihood estimator (MLE) of $\Theta$, we want to maximize the log-likelihood function

$$\log L(\Theta) = \log \left[ \prod_{i=1}^n f(x_i; \Theta) \right] = \sum_{i=1}^n \log f(x_i; \Theta) = \sum_{i=1}^n \log \left[ \sum_{k=1}^K \pi_k f_k(x_i; \theta_k) \right]$$

with respect to $\Theta$. In general, there is no closed form for the maximum likelihood estimates due to numerical difficulty coming from the sum inside the log. The expectation-maximization (EM) algorithm introduced by Dempster et al. [6] can be applied by casting the problem in the framework of missing data. Under our model assumptions, the complete data are considered to be $(x_i, z_i)$, where $z_i = (z_{i1}, z_{i2}, ..., z_{ik})$ is the unobserved latent variable, with

$$z_{ik} = \begin{cases} 1 & \text{if } x_i \text{ is from component } k; \\ 0 & \text{otherwise.} \end{cases}$$

Assuming that $z_i$'s are iid, following a multinomial distribution with $K$ categories and probabilities $\pi_1, \pi_2, ..., \pi_K$, the density of an observation $x_i$ given $z_i$ is then

$$f(x_i; \Theta | z_i) = \prod_{k=1}^K f_k(x_i; \theta_k)^{z_{ik}}.$$ 

Therefore the density of the complete data is
\[ f(x_i; \Theta) = f(x_i; \Theta | z_i) f(z_i) = \prod_{k=1}^{K} \left[ f_k(x_i; \theta_k) \right]^{z_{ik} \pi_k} = \prod_{k=1}^{K} \left[ \pi_k f_k(x_i; \theta_k) \right]^{z_{ik}}. \]

The resulting log-likelihood for the complete data can be represented as

\[
\log L_{c}(\Theta) = \sum_{i=1}^{n} \log f(x_i; \Theta) = \sum_{i=1}^{n} \log \left[ \prod_{k=1}^{K} \left[ \pi_k f_k(x_i; \theta_k) \right]^{z_{ik}} \right] = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \left[ \log \pi_k + \log f_k(x_i; \theta_k) \right] \\
= \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \left[ \log \pi_k - \frac{P}{2} \log (2\pi) - \frac{1}{2} \sum_{p=1}^{p} \log \sigma_p^2 - \frac{1}{2} \sum_{p=1}^{p} \left( x_{ip} - \mu_{kp} \right)^2 \right].
\]

The EM algorithm seeks to find the maximum likelihood estimate

\[ \hat{\Theta} = \left\{ \hat{\pi}, \hat{\mu}_{kp}, \hat{\sigma}_k^2 : k = 1, \ldots, K; p = 1, \ldots, P \right\} \]

by iteratively applying the following two steps:

**Expectation step** (E-step): compute the expected value of the above complete log-likelihood function (1) with respect to the conditional distribution of \( z \) given the data \( X \) and the estimate \( \hat{\Theta}^{(r)} \) of \( \Theta \) at the \( r \)-th iteration:

\[
Q(\Theta | \hat{\Theta}^{(r)}) = E_{z_i|X, \hat{\Theta}^{(r)}} \left( \log L_{c}(\Theta) \right) \\
= \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \left[ \log \pi_k + \log f_k(x_i; \theta_k) \right] \hat{P}(z_{ik} = 1 | x_i) + \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \left[ \log \pi_k + \log f_k(x_i; \theta_k) \right] \hat{P}(z_{ik} = 0 | x_i) \\
= \sum_{i=1}^{n} \sum_{k=1}^{K} \left[ \log \pi_k + \log f_k(x_i; \theta_k) \right] \hat{P}(z_{ik} = 1 | x_i).
\]

Set \( \hat{z}_{ik}^{(r)} = \hat{P}(z_{ik} = 1 | x_i) \) to be the posterior probability that \( x_i \) comes from the \( k \)-th component, computed through Bayes’ theorem at the \( r \)-th iteration:

\[
\hat{z}_{ik}^{(r)} = \hat{P}(z_{ik} = 1 | x_i) = \frac{\hat{P}(z_{ik} = 1, x_i) \hat{P}(x_i)}{\hat{P}(x_i)} = \frac{\hat{P}(z_{ik} = 1) \hat{P}(x_i | z_{ik} = 1)}{\sum_{k=1}^{K} \hat{P}(z_{ik} = 1) \hat{P}(x_i | z_{ik} = 1)} = \frac{\hat{\pi}_k f_k(x_i; \hat{\Theta}^{(r)}) \hat{z}_{ik}^{(r)}}{\sum_{k=1}^{K} \hat{\pi}_k f_k(x_i; \hat{\Theta}^{(r)})}.
\]

We have
\[ Q(\Theta | \hat{\Theta}^{(r)}) = \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{z}_{ik}^{(r)} \left[ \log \pi_k + \log f_k (x_i; \theta_k) \right] \]
\[ = \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{z}_{ik}^{(r)} \left[ \log \pi_k - \frac{P}{2} \log (2\pi) - \frac{1}{2} \sum_{p=1}^{p} \sigma_p^2 - \frac{1}{2} \sum_{p=1}^{p} \left( x_{ip} - \mu_{kp} \right)^2 \right]. \tag{3} \]

**Maximization step** (M-step): maximize the above quantity (3) in terms of \( \pi_k \) and \( \theta_k \) with \( \hat{z}_{ik} \) being fixed at the values computed in the E-step: \( \hat{\Theta}^{(r+1)} = \arg \max_{\Theta} Q(\Theta | \hat{\Theta}^{(r)}) \). It is not difficult to show that

\[ \hat{\pi}_k^{(r)} = \frac{\sum_{i=1}^{n} \hat{z}_{ik}^{(r)}}{n}, \]
\[ \hat{\mu}_{kp}^{(r+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ik}^{(r)} x_{ip}}{\sum_{i=1}^{n} \hat{z}_{ik}^{(r)}}, \]
\[ \hat{\sigma}_p^{2(r+1)} = \frac{\sum_{i=1}^{n} \sum_{k=1}^{K} \hat{z}_{ik}^{(r)} \left( x_{ip} - \hat{\mu}_{kp}^{(r)} \right)^2}{n}. \]

Starting from initial values \( \Theta^{(0)} = \{ \pi_k^{(0)}, \mu_{kp}^{(0)}, \sigma_p^{2(0)} : k = 1,...,K; p = 1,...,P \} \) obtained from the K-means algorithm \(^7\), repeat the above iterations until convergence. Then the posterior probability of any observation belonging to each cluster can be calculated from (2).

### 2.2 Penalized model-based clustering

With the same motivation as in penalized regression for variable selection, Pan and Shen proposed a penalized model-based clustering approach with an \( L_1 \) penalty. Specifically, they regularize \( \log L(\Theta) \) by adding a penalty function \( p_{\lambda}(\Theta) \) with penalization parameter \( \lambda \), yielding a penalized log-likelihood function for the original data

\[ \log L_p (\Theta) = \sum_{i=1}^{n} \log \left[ \sum_{k=1}^{K} \pi_k f_k (x_i; \theta_k) \right] - p_{\lambda}(\Theta). \]
The corresponding penalized log-likelihood function for the complete data is

\[
\log L_{c,p}(\Theta) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} \left[ \log \pi_k + \log f_k(x_i; \theta_k) \right] - p_k(\Theta).
\]

The form of \(p_k(\Theta)\) depends on the goal of the analysis. In order to shrink the means of non-informative components towards the global mean, Pan and Shen used the \(L_1\) penalty

\[
p_k(\Theta) = \lambda \sum_{k=1}^{P} \sum_{p=1}^{P} |\mu_{kp}|,
\]

where \(\mu_{kp}\) is the \(p\)-th component of \(\mu_k\). Before clustering analysis is done, standardize the data so that each attribute has sample mean 0 and sample variance 1. Then the goal of choosing an appropriate \(L_1\) penalty is to coerce small estimates of \(\mu_{kp}\) to be exactly 0, and thus realize the identification of non-informative attributes. For example, if we obtain \(\mu_{1p} = \mu_{2p} = \ldots = \mu_{kp} = 0\), then the \(p\)-th variable will be treated as a non-informative variable which only contributes noise to the clustering analysis.

Under the same assumption of normal mixtures, the EM algorithm for parameter estimation with the above penalized model-based clustering can be derived in a similar fashion. Pan and Shen obtained the following results after some algebraic manipulations:

\[
Q(\Theta | \Theta^{(r)}) = E_{z|x,\hat{\Theta}^{(r)}}(\log L_{c,p}(\Theta))
\]

\[
= \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{z}_{ik}^{(r)} \left[ \log \pi_k + \log f_k(x_i; \theta_k) \right] - \lambda \sum_{k=1}^{K} \sum_{p=1}^{P} |\mu_{kp}|
\]

\[
= \sum_{i=1}^{n} \sum_{k=1}^{K} \hat{z}_{ik}^{(r)} \left[ \log \pi_k - \frac{P}{2} \log (2\pi) - \frac{1}{2} \sum_{p=1}^{P} \log \sigma_p^2 - \frac{1}{2} \sum_{p=1}^{P} \left( x_{ip} - \mu_{kp} \right)^2 \sigma_p^2 \right] - \lambda \sum_{k=1}^{K} \sum_{p=1}^{P} |\mu_{kp}|
\]

Initialize parameters \(\Theta^{(0)} = \{\pi_k^{(0)}, \mu_{kp}^{(0)}, \sigma_p^{2(0)} : k = 1, \ldots, K; p = 1, \ldots, P\}\) using K-means, and then in the E-step, calculate

\[
\hat{z}_{ik}^{(r)} = \frac{\hat{\pi}_k^{(r)} f_k(x_i; \hat{\theta}_k^{(r)})}{\sum_{k=1}^{K} \hat{\pi}_k^{(r)} f_k(x_i; \hat{\theta}_k^{(r)})}.
\]
In the M-step, update with

\[ \hat{\pi}_k^{(r)} = \frac{\sum_{i=1}^n \hat{x}_{ik}^{(r)}}{n}, \]

\[ \hat{\mu}_{kp}^{(r+1)} = \hat{\mu}_{kp}^{(r)} \left( 1 - \frac{\lambda_p \hat{\sigma}_p^{2,r}}{n \sum_{i=1}^n \hat{x}_{ik}^{(r)} x_{ip}^2} \right) \]

\[ \hat{\sigma}_p^{2,(r+1)} = \frac{\sum_{i=1}^n \tau_{ik}^{r+1} x_{ip}^2}{\sum_{i=1}^n \tau_{ik}^{r+1}}, \]

where \( f_+ = \begin{cases} f & \text{if } f > 0; \\ 0 & \text{otherwise}. \end{cases} \). Repeat the above steps until convergence and calculate the posterior probabilities.

### 2.3 Gap statistics

Using the same notation for our observed data \( x_i = \{x_{ip} : p = 1, 2, ..., P\}, i = 1, 2, ..., n \), the idea of Gap statistics proposed by Tibshirani et al. is as follows:

Denote the distance between observation \( i \) and \( i' \) by \( d_{i'i} \), where \( d_{i'i} = \sum_i (x_{ip} - x_{ip'})^2 \) under Euclidean distance.

Suppose we cluster the observations into \( K \) clusters \( C_1, C_2, ..., C_K \), with \( n_k = |C_k| \) which is the number of observations in \( C_k \). Let

\[ D_k = \sum_{j, f \in C_k} d_{j'f} \]

be the sum of pairwise distances for all observations in cluster \( k \) and

\[ W_K = \sum_{k=1}^K \frac{1}{2n_k} D_k \tag{4} \]

be the pooled within cluster sum of squares around the cluster means.
Gap statistic is defined as

\[
Gap_n(K) = E_n^* \left( \log(W_K) \right) - \log(W_K),
\]

where \( E_n^* \) denotes expectation with a sample of size \( n \) from the reference distribution generated by Monte Carlo. We adopt the choice of uniform distribution as the reference distribution and generate the reference features over a box aligned with the principal components of the data. Tibshirani summarized the computational implementation of the Gap statistic as follows:

Cluster the observed data using K-means with various numbers of clusters \( K = 1, 2, \ldots, M \), where \( M \) is a given number. Calculate \( W_K \) by (4).

Generate \( B \) sets of simulated data using the uniform distribution. Cluster each one and calculate \( W^{*}_{Kb} \), \( b = 1, 2, \ldots, B, K = 1, 2, \ldots, M \). Compute the estimated Gap statistics

\[
Gap(K) = (1/B) \sum_b \log(W^{*}_{Kb}) - \log(W_K).
\]

Let \( \bar{I} = (1/B) \sum_b \log(W^{*}_{Kb}) \), calculate \( sd_K = \left[ (1/B) \sum_b \left( \log(W^{*}_{Kb}) - \bar{I} \right)^2 \right]^{1/2} \), and define

\[
s_K = sd_K \sqrt{1 + 1/B}.
\]

Choose the optimal number of clusters \( \hat{K} \) where \( \hat{K} \) is the smallest \( K \) such that

\[
Gap(K) \geq Gap(K + 1) - s_{K+1}.
\]

2.4 Model selection

In Pan and Shen’s grid search approach, both the number of clusters \( K \) and penalization parameter \( \lambda \) need to be estimated. By applying Gap statistics prior to the penalized clustering analysis, we are able to fix \( K \), and choose the optimal \( \lambda \) according to Bayesian Information Criterion (BIC) or other criteria.
Following Pan and Shen, we use their modified BIC as our model selection criterion. It is defined as

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

where \( d_e = K + P + KP - 1 - q \) and \( q = \#\{(k, p): \mu_{kp} = 0\} \). Here \( d_e \) is the effective number of unknown parameters; \( q \) is the number of non-informative variables. After obtaining the estimated \( K \) via Gap statistics, we specify several proposed values of \( \lambda \) for the penalized likelihood clustering step. For each given \( \lambda \), we run the EM algorithm multiply times with randomly started K-means’ results as initial values, and select the one with the maximal penalized log-likelihood. We choose the \( \lambda \) which produces the minimum BIC and use it for model evaluation.

In the following two chapters, we evaluate our method with both simulated data and real gene expression data. We start from simulation studies in Chapter 3, and then provide two real data examples in Chapter 4 to demonstrate the usefulness of our approach.
3 Simulations

As the non-informative variables can mask the true cluster structure by introducing noise into the analysis, they can also lead us to incorrect estimation of $K$, the number of components. To investigate the impact of noise variables and the robustness of our method, we simulated two “high dimension, low sample size” datasets with a large proportion of their variables being non-informative, and evaluated our approach by clustering result and identification of the non-informative variables.

3.1 Simulation set-ups

In each simulated dataset, there are $P = 270$ variables with $n = 90$ observations. Only the first 20 variables are informative while the other 250 are noise variables. For set-up 1 (two-true-cluster case), the 90 observations are simulated from $K = 2$ clusters based on the 20 effective variables, with 44 observations in one cluster and 46 in the other. The 20 informative variables are iid from $N(5,1)$ for the first cluster, whereas they are iid from $N(6.5,1)$ for the second cluster. The non-informative variables are generated by iid $N(0,1)$.

For set-up 2 (three-true-cluster case), the 90 observations are generated from $K = 3$ clusters, with 30 observations in each cluster. The informative variables come from $N(5,1)$, $N(6.5,1)$, and $N(8,1)$ for the three clusters respectively.

3.2 Results

With both set-ups, Gap statistics was able to determine the correct number of clusters $K$, even when a large amount of noise variables existed. Using the obtained $K$ value, we chose various penalization parameters $\lambda = 1, 5, 6, 7, 8, 9, 10, 15,$ and $20$ for clustering analysis with penalized likelihood. For each $\lambda$, the modified BIC, clustering result, and number of noise variables detected, $r$, were calculated. Results are shown in Tables 1 and 2.
Table 1  Simulation results with set-up 1. The truth was $K = 2$, $r = 250$.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>1</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>$BIC$</td>
<td>71135</td>
<td>70281</td>
<td>70115</td>
<td>70042</td>
<td>69999</td>
<td>69881</td>
<td>69885</td>
<td>69960</td>
<td>70085</td>
</tr>
<tr>
<td>$q$</td>
<td>34</td>
<td>167</td>
<td>192</td>
<td>206</td>
<td>216</td>
<td>234</td>
<td>238</td>
<td>249</td>
<td>250</td>
</tr>
</tbody>
</table>

Table 2  Simulation results with set-up 2. The truth was $K = 3$, $r = 250$.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>1</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>15</th>
<th>20</th>
</tr>
</thead>
<tbody>
<tr>
<td>$BIC$</td>
<td>71095</td>
<td>69799</td>
<td>69644</td>
<td>69465</td>
<td>69377</td>
<td>69352</td>
<td>69358</td>
<td>69565</td>
<td>69843</td>
</tr>
<tr>
<td>$q$</td>
<td>10</td>
<td>124</td>
<td>151</td>
<td>183</td>
<td>206</td>
<td>220</td>
<td>231</td>
<td>249</td>
<td>250</td>
</tr>
</tbody>
</table>

Figure 1  Modified BICs for $\lambda = 1, 5, 6, 7, 8, 9, 10, 15, 20$ with set-up 1, $K = 2$
Since Gap statistics was able to obtain the true $K$ value, we used the clustering result to directly evaluate our models. For both set-ups, every value $\lambda$ has provided the correct clustering result: $\hat{C} = (1,1,\ldots,1,2,2,\ldots,2)$ for set-up 1, and $\hat{C} = (1,1,\ldots,1,2,2,\ldots,2,3,3,\ldots,3)$ for set-up 2.

Figure 1 and 2 plot the modified BICs corresponding to different choices of $\lambda$. If we choose the $\lambda$ with the lowest BIC, we are able to detect 234 of the 250 noise variables for set-up 1 ($\lambda = 9$), and 220 of 250 for set-up 2 ($\lambda = 9$). Recall that we treat a variable as noise if all of its mean estimates are exactly equal to zero.
Therefore, as long as we have the correct number of clusters, penalized model-based clustering has very good performance on selecting the non-informative variables. By forcing the mean estimates to be zero, the model fit can be also improved, as shown in Pan and Shen’s work.
4 Examples

Modern biotechnologies such as microarrays normally produce high-dimensional data. Clustering analysis has become a powerful tool in handling this type of data. It can be used to classify or discover gene functions, disease (cancer) subtypes, etc. In these tasks, variable selection is an important step, because it is known that not all of the genes are associated with the disease or any phenotype of interest. Microarray data may contain expression levels of thousands of genes, but only a proportion of them will be informative to clustering analysis. Successfully eliminating the non-informative variables through variable selection not only provides valuable biological information, but also improves the performance of clustering.

In this chapter, we applied our method to two real data examples. In both examples we know the true number of clusters, so we can justify the performance of Gap statistics. However, we don’t have additional information on the selection of informative variables.

4.1 Novartis multi-tissue data

The Novartis multi-tissue data is from Su et al. [8] on profiling gene expression from mouse and human tissue samples. The tissue samples used in their experiment were from four distinct cancer types: 26 breast cancer (BR), 26 prostate cancer (PR), 28 lung cancer (LU), and 23 colon cancer (CO). Gene expression levels of 1000 genes were profiled for each tissue sample. See Table 3 for the structure of the Novartis multi-tissue data.

Suppose we did not know which cancer type each tissue sample was associated with, nor the number of cancer types, and were interested in both cancer type discovery and relevant-gene selection, we could apply our approach to this data, and then evaluate our result with the true cancer type for each tissue sample.

Since the original data takes hours for a single run, we only used a subset of it for illustration. We picked the first 6 samples from each cancer type, and the first 100 of the 1000 genes. Therefore the truth was $K = 4$, $n = 24$, and $P = 100$. 
Table 3  Sample Novartis multi-tissue data

<table>
<thead>
<tr>
<th>Tissue sample (observations)</th>
<th>Gene index (variables)</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>...</td>
<td>1000</td>
</tr>
<tr>
<td>BR1</td>
<td>-276.29</td>
<td>173.08</td>
<td>131.16</td>
<td>...</td>
<td>63.43</td>
<td></td>
</tr>
<tr>
<td>BR2</td>
<td>633.2</td>
<td>49.71</td>
<td>119.49</td>
<td>...</td>
<td>12.21</td>
<td></td>
</tr>
<tr>
<td>BR3</td>
<td>295.43</td>
<td>219.1</td>
<td>85.26</td>
<td>...</td>
<td>67.41</td>
<td></td>
</tr>
<tr>
<td>BR26</td>
<td>91.9</td>
<td>134.32</td>
<td>411.03</td>
<td>...</td>
<td>47.47</td>
<td></td>
</tr>
<tr>
<td>PR1</td>
<td>943.68</td>
<td>27.81</td>
<td>91.11</td>
<td>...</td>
<td>4.8</td>
<td></td>
</tr>
<tr>
<td>PR26</td>
<td>1954.77</td>
<td>38.1</td>
<td>83.04</td>
<td>...</td>
<td>-7.82</td>
<td></td>
</tr>
<tr>
<td>CO23</td>
<td>-91.82</td>
<td>131.18</td>
<td>282.97</td>
<td>...</td>
<td>-24.36</td>
<td></td>
</tr>
</tbody>
</table>

With our approach, Gap statistics successfully selected the true value $K = 4$. Then we used candidates of $\lambda = 1, 2, 3, 4, \text{ and } 5$, and calculated the corresponding BICs, shown in Figure 3. $\lambda = 4$ produced the smallest BIC, and therefore was chosen for variable selection. 5 genes have their mean estimates equal to zero across all 4 clusters, and therefore were identified as non-informative.

For all the values of $\lambda$ we used, our approach was able to capture the correct clustering structure of the data: $\hat{C} = (1,1,1,1,1,2,2,2,2,2,2,2,2,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4)$. Thus the potential non-informative genes have not masked the true clustering structure of the data.
in this example. However, this situation may change if we use the original dataset with all 1000 genes included.

Figure 3  Modified BICs for $\lambda = 1, 2, 3, 4, 5$ with Novartis multi-tissue data

4.2 Lung cancer data

Another real data example is the lung cancer data [9] by Bhattacharjee et al. They analyzed mRNA expression levels in lung tumor samples from 4 known classes of lung cancer: 139 adenocarcinomas (AD) tumor samples, 21 squamous cell carcinomas (SQ), 20 carcinoids (COID), and 17 normal lung (NL). The data structure is displayed in Table 4. We used a subset of the original dataset with the first 6 observations from each class, and the first 200 of the 1000 genes. The truth was $K = 4$, $n = 24$, and $P = 200$ in this case.
Table 4  Sample lung cancer data

<table>
<thead>
<tr>
<th>Tumor sample (observations)</th>
<th>Gene index (variables)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td>AD1</td>
<td>33.21</td>
</tr>
<tr>
<td>AD139</td>
<td>-23.79</td>
</tr>
<tr>
<td>NL1</td>
<td>59.09</td>
</tr>
<tr>
<td>NL17</td>
<td>74.79</td>
</tr>
<tr>
<td>COID20</td>
<td>-13.31</td>
</tr>
</tbody>
</table>

Figure 4  Modified BICs for $\lambda = 1, 2, 3, 4, 5$ with lung cancer data
Again, Gap statistics has provided the correct number of clusters. Using the same candidates for $\lambda$ as in the previous analysis, we obtained the BICs shown in Figure 4. We chose $\lambda = 2$ as the optimal penalization parameter. Although many mean estimates are exactly zero, no gene has zero estimates across all 4 clusters. Thus we conclude that all the 200 genes have provided some information in the analysis. The clustering result with our choice of $K$ and $\lambda$ is $\hat{C} = (1,1,1,1,2,2,2,2,3,3,3,3,4,4,4,4,4,4)$. 
5 Discussion

From our simulation results and real data analysis, we found Gap statistics to be very effective in capturing the correct number of clusters $K$ even if a large proportion of noise variables are present. After the estimation of the appropriate number of clusters is done via Gap statistics, the computational cost for clustering analysis with penalized normal mixture model is greatly reduced because only an optimal penalization parameter $\lambda$ needs to be selected through grid search.

Moreover, since the grid search approach requires a prior specification of the candidates for both $K$ and $\lambda$, and the BIC depends on both of them, failure to include the true value of $K$ in its candidates or a poor estimation of $\lambda$ can result in an incorrect estimation of $K$, which is more serious than a non-optimal choice of $\lambda$. In our simulation studies, as long as we have the correct estimate of $K$, different values of $\lambda$ produced the same and correct clustering results. Only the identification of the non-informative variables is affected by $\lambda$ since it controls the shrinkage of the cluster means.

For simplicity we have used an equal diagonal covariance matrix across clusters in our method. Results from real data analysis have showed good performance of our method. However, the assumption of common covariance matrix across different clusters is not easy to meet in many problems. Therefore a cluster-specific diagonal or even unconstrained covariance matrix should be assumed, as in Xie et al., but it evolves more parameters to estimate and more computational power to implement the algorithm.
References


Appendix

A.1 R code for determining number of clusters via Gap statistics

```r
library(SAGx)

indata <- read.csv("D:/Rsimulation/tempdata2.csv",header=FALSE,sep="","

B <- 50
K <- 10
gp <- matrix(NA,ncol=1,nrow=K)
sd <- matrix(NA,ncol=1,nrow=K)
for(k in 2:K){
g <- matrix(NA,ncol=1,nrow=B)
for (i in 1:B){
cl <- myclus(data = indata,k)
g[i]<- gap(indata,cl$cluster,1)
}
gp[k] <- mean(g)
sd[k] <- sqrt(1+1/B)*(mean((g-mean(g))^2))^(1/2)
}
gp
ds
min(c(2:9)[res >= 0],na.rm=T)
```

A.2 R code for penalized model-based clustering

```r
library(mvtnorm)
library(mclust)

lambda <- 20
K <- 3
delta <- 0.01

indata <- read.csv("D:/Rsimulation/tempdata2.csv",header=FALSE,sep="","

indata <- apply(indata,2,function(x) (x-mean(x))/sd(x))
n <- nrow(indata)
P <- ncol(indata)
kmeans.result <- kmeans(indata, centers=K)
pr <- c(1:K)
for (k in 1:K){
pr[k]=sum(kmeans.result$cluster==k)/n
}
theta <- list(
    pr=pr,
    mu=kmeans.result$centers,
    sig=diag(1,P)
)

Tau <- function(theta){
cbnd = theta$pr[1]*dmvnorm(indata,mean=theta$mu[1,],sigma=theta$Sig)
for (k in 2:K){
cbnd=cbind(cbnd,theta$pr[k]*dmvnorm(indata,mean=theta$mu[k,],sigma=theta$Sig))
}
```
t(apply(cbnd,1,function(x) x/sum(x)))
}

den <- numeric()
for (k in 1:K){
    den[k]=sum(Tau(theta) [,k])
}

num <- matrix(c(1:(K*P)),K,P)
for (k in 1:K){
    for(p in 1:P){
        num[k,p]=sum(Tau(theta) [,k]*indata[,p])
    }
}

noneg <- matrix(c(1:(K*P)),K,P)
for(k in 1:K){
    for(p in 1:P){
        noneg[k,p]=1-lambda*theta$sig[p,p]/abs(num[k,p])
    }
}

mu <- matrix(c(1:(K*P)),K,P)
for(k in 1:K){
    for(p in 1:P){
        mu[k,p]=num[k,p]/den[k]*ifelse(noneg[k,p]>0,noneg[k,p],0)
    }
}

sigp <- numeric()
for (p in 1:P){
    temp = 0*c(1:n)
    for (k in 1:K){
        temp = temp + Tau(theta) [,k]*((indata[,p]-theta$mu[k,p])^2)
    }
    sigp[p] = sum(temp)/n
}

theta_updates <- function(T) list(
    pr=apply(T,2,mean),
    mu=mu,
    sig=diag(sigp,P)
)

temp = 0
for (k in 1:K){
    temp = temp +
    theta$pr[k]*dmvnorm(indata,mean=theta$mu[k,],sigma=theta$sig)
}
likelihood <- sum(log(temp))-lambda*sum(abs(theta$mu))
q <- 0
dimension <- K+P+K*P-1-q
BIC <- -2*likelihood+log(n)*dimension
BICT <- 0
while(abs(BICT-BIC)>delta){
    BICT <- BIC
    T <- Tau(theta)
    den <- numeric()
for (k in 1:K) {
    den[k] = sum(Tau(theta)[,k])
}

num <- matrix(c(1:(K*P)), K, P)
for (k in 1:K) {
    for (p in 1:P) {
        num[k, p] = sum(Tau(theta)[,k] * indata[,p])
    }
}

noneg <- matrix(c(1:(K*P)), K, P)
for (k in 1:K) {
    for (p in 1:P) {
        noneg[k, p] = 1 - lambda*theta$sig[p,p]/abs(num[k,p])
    }
}

mu <- matrix(c(1:(K*P)), K, P)
for (k in 1:K) {
    for (p in 1:P) {
        mu[k, p] = num[k, p]/den[k]*ifelse(noneg[k,p]>0, noneg[k,p], 0)
    }
}

sigp <- numeric()
for (p in 1:P) {
    temp = 0*c(1:n)
    for (k in 1:K) {
        temp = temp + Tau(theta)[,k]*((indata[,p]-theta$mu[k,p])^2)
    }
    sigp[p] = sum(temp)/n
}

theta <- theta_updates(T)

temp = 0
for (k in 1:K) {
    temp = temp + theta$pr[k]*dmvnorm(indata, mean=theta$mu[k,], sigma=theta$sig)
}
likelihood <- sum(log(temp))-lambda*sum(abs(theta$mu))

q <- length(which(theta$mu==0))
dimension <- K+P*K*P-1-q
BIC <- -2*likelihood+log(n)*dimension

cluster <- numeric()
for (i in 1:n) {
    cluster[i]=which(Tau(theta)[i]==max(Tau(theta)[i]))
}

cluster
BIC

A.3  R code for noise variables selection with

P <- length(mu[1,])
optime <- matrix(c(1:K), K, 1)
for (p in 1:P) {
if(sum(abs(mu[,p]))==0){
    noise <- cbind(noise,mu[,p])
}
noise <- noise[,,-1]

newdata <- indata
i=0
for (p in 1:P){
    if(sum(abs(mu[,p]))==0){
        newdata <- newdata[,,-p+i]
        i=i+1
    }
}
NP <- ncol(newdata)
ncol(noise)