Biometrics 64, 440–448
June 2008

DOI: 10.1111/j.1541-0420.2007.00922.x

Variable Selection for Model-Based High-Dimensional Clustering and Its Application to Microarray Data

Sijian Wang[1] **and Ji Zhu**[2,*]

[1]Department of Biostatistics, University of Michigan, Ann Arbor, Michigan 48109, U.S.A.
[2]Department of Statistics, University of Michigan, Ann Arbor, Michigan 48109, U.S.A.
[*]*email:* jizhu@umich.edu

Summary. Variable selection in high-dimensional clustering analysis is an important yet challenging problem. In this article, we propose two methods that simultaneously separate data points into similar clusters and select informative variables that contribute to the clustering. Our methods are in the framework of penalized model-based clustering. Unlike the classical L_1-norm penalization, the penalty terms that we propose make use of the fact that parameters belonging to one variable should be treated as a natural "group." Numerical results indicate that the two new methods tend to remove noninformative variables more effectively and provide better clustering results than the L_1-norm approach.

Key words: EM algorithm; High-dimension low sample size; Microarray; Model-based clustering; Regularization; Variable selection.

1. Introduction

Clustering data into similar clusters is an important practical problem in a wide variety of fields, including statistics, bioinformatics, artificial intelligence, and data mining. With the recent advent of technologies, good clustering algorithms are very much desired for analyzing high-dimensional data where the number of variables is considerably larger than the number of observations. DNA microarray analysis is a typical example that involves such high-dimensional data. A microarray data set usually has thousands or tens of thousands of gene expression profiles (variables), but only around tens or hundreds of samples. Clustering microarray data can be very helpful for certain types of biology study, such as cancer research. For example, based on the gene expression profiles, interesting cluster distinctions can be found among a set of tissue samples, which may reflect categories of diseases, mutational status, or different responses to a certain drug. Besides separating samples into distinct clusters, another challenge in microarray analysis is to identify the informative genes that contribute most to the clustering. This is a variable selection problem.

Variable selection has been studied extensively in the literature for regression and classification problems (Breiman, 1995; Tibshirani, 1996; Fan and Li, 2001; Zhao, Rocha, and Yu, 2006; Zou and Yuan, 2006), but not so much for clustering. However, selecting informative variables and removing noninformative variables are also important for clustering. Figure 1 illustrates the point.

In Figure 1, we can easily separate points into two clusters by visual inspection. We also observe that variable x_2 does not contribute to the cluster discrimination. In fact, if we use x_2 alone, there will only be a single cluster, which is less interesting. This demonstrates that noninformative variables can mislead clustering results. The problem can be more severe when the number of noninformative variables is large, where the noninformative variables can "hide" the true cluster structure.

Pan and Shen (2006) proposed an approach for variable selection in clustering through penalized model-based clustering. Following Liu, Zhang, and Palumbo (2003) and Hoff (2006), they parameterized the mean in cluster k for variable x_j as $\mu_{kj} = \phi_j + \delta_{kj}$, where ϕ_j is the global mean for variable x_j. If for different k, all δ_{kj} are 0, then the variable x_j is not informative for clustering, at least in terms of the mean. Pan and Shen (2006) employed the L_1-norm penalty to shrink the cluster-specific means μ_{kj} toward the global mean ϕ_j, and this effectively realizes the variable selection.

In this article, we focus on the clustering for high-dimensional data characterized by high dimension and low sample size (Marron and Todd, 2002). Enlightened by the method in Pan and Shen (2006), we propose two new approaches that are also in the framework of penalized model-based clustering. Noting that cluster-specific mean parameters associated with the same variable can be naturally "grouped" together, and intuitively should be treated as a "group," we propose two novel penalty functions, different from the one in Pan and Shen (2006), to make use of such natural "grouping" information within the data. As we will see in the numerical study, the two new methods tend to remove noninformative variables more effectively and provide better clustering results.

There are several other methods in the literature that combine clustering and variable selection together. Friedman and Meulman (2004) proposed a hierarchical clustering procedure that uncovers cluster structure on separate subsets of variables. The algorithm does not explicitly select variables but rather assigns them different weights, which can be used to extract informative variables. Analogous to stepwise variable selection in regression, Raftery and Dean (2006) developed a method to sequentially compare two nested models to determine whether a subset of variables should be included in or excluded from the current model based on a greedy search. Hoff (2006) proposed a multivariate Dirichlet mixture process based on a Pólya urn cluster model for multivariate means and variances. This approach finds clusters that differ from each other in terms of their means and/or variances on one or more variables. Tadesse, Sha, and Vannucci (2005) formulated the clustering problem in terms of a multivariate normal mixture model with an unknown number of components. They used the reversible jump Markov chain Monte Carlo technique to define a sampler that could move between different dimensional spaces, and the variable selection was handled through the introduction of a binary exclusion/inclusion latent vector.

An alternative strategy for high-dimensional clustering is to first apply certain dimension reduction methods on the data, for example, principal component analysis (PCA) or correspondence analysis (CA), then apply clustering on the reduced space. However, treating dimension reduction and clustering as two separate steps may destroy the cluster structure in the data, as pointed out in Raftery (2003). For example, in Figure 1, the first principal component may be x_2, which is noninformative for discovering the cluster structure. Liu et al. (2003) proposed a Bayesian clustering procedure after transforming the data via the PCA. The number of principal components to be included and which principal components should be included are automatically selected using Gibbs sampling. However, the extracted components usually are linear combinations of *all* the original variables, thus there is no variable selection.

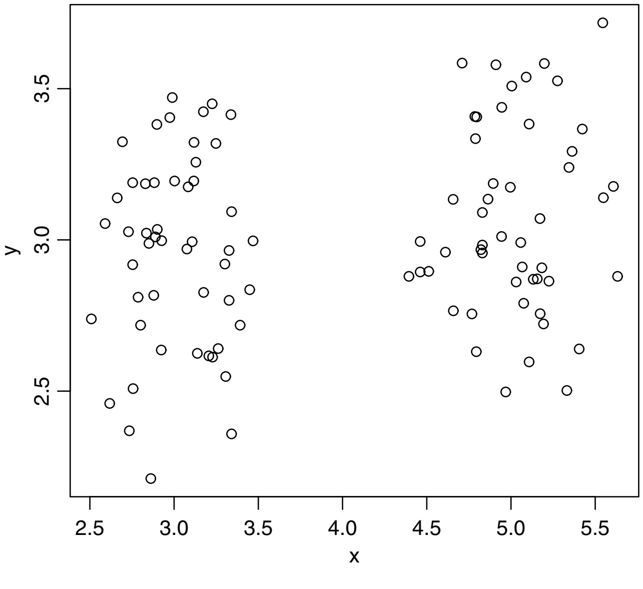

Figure 1. An illustration of the noninformative variable in clustering analysis. In this example, variable x_2 does not contribute to the cluster discrimination, and we consider it as noninformative.

The rest of the article is organized as follows. In Section 2, we propose our two new models: the adaptive L_∞-norm penalized Gaussian mixture model (ALP-GMM), and the adaptive hierarchically penalized Gaussian mixture model (AHP-GMM). In Section 3, we derive algorithms to estimate the parameters in the two models. Numerical results are in Sections 4 and 5. We conclude the article with Section 6.

2. Models

In this section, we propose two variable selection methods for clustering high-dimensional data.

We observe n p-dimensional samples $\boldsymbol{x}_i = (x_{i1}, \ldots, x_{ij}, \ldots, x_{ip})$, $i = 1, \ldots, n$, and without loss of generality, we assume that the data are centered in each dimension (variable), that is, $\sum_{i=1}^n x_{ij} = 0, j = 1, \ldots, p$. Our aim is to separate the data into K clusters.

The Gaussian mixture model (GMM) is a standard tool for this purpose (Fraley and Raftery, 2002; McLachlan and Peel, 2002). We assume that each observation $\boldsymbol{x}_i$ is drawn from a finite Gaussian mixture distribution:

$$f(\boldsymbol{x}_i) = \sum_{k=1}^K \pi_k f_k(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k),$$

where π_k's are the mixing proportions satisfying $0 \leq \pi_k \leq 1$ and $\sum_{k=1}^K \pi_k = 1$. $\boldsymbol{\mu}_k = (\mu_{k1}, \ldots, \mu_{kj}, \ldots, \mu_{kp})$ is the mean vector of the Gaussian distribution characterizing the kth cluster, and $\boldsymbol{\Sigma}_k$ is the corresponding covariance matrix. In this article, we focus on high-dimensional data and assume $\boldsymbol{\Sigma}_k = \boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \ldots, \sigma_j^2, \ldots, \sigma_p^2)$, that is, the covariance matrices are the same across different clusters and are diagonal. This is a common assumption when one works with high dimension and low sample size data, for example, in the naive Bayes model. Some theoretical justification for this assumption in the case of discriminant analysis can be found in Bickel and Levina (2004).

Given an observation $\boldsymbol{x}^* = (x_1^*, \ldots, x_p^*)$, one can compute the probability that $\boldsymbol{x}^*$ is from the kth cluster

$$p_k = \frac{\pi_k}{\sqrt{2\pi}\prod_{j=1}^p \sigma_j} \exp\left(-\sum_{j=1}^p \frac{\left(x_j^* - \mu_{kj}\right)^2}{2\sigma_j^2}\right), \quad k = 1, \ldots, K \tag{1}$$

and $\boldsymbol{x}^*$ will be assigned to the cluster with the largest p_k.

We denote $\Theta = \{\sigma_j^2, \pi_k, \mu_{kj}, k = 1, \ldots, K; j = 1, \ldots, p\}$ as the set containing all the parameters. Given the data $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$, the log-likelihood function is

$$\ell_0(\Theta) = \sum_{i=1}^n \log\left(\sum_{k=1}^K \pi_k f_k(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})\right). \tag{2}$$

Maximization of the above objective function with respect to Θ is often difficult, and it is common to use the expectation maximization (EM) algorithm (Dempster, Laird, and Rubin, 1977) via the framework of missing data. Let τ_{ik} be the indicator of whether $\boldsymbol{x}_i$ is from cluster k, that is, $\tau_{ik} = 1$ if $\boldsymbol{x}_i$ belongs to cluster k, and $\tau_{ik} = 0$ otherwise. If the missing data τ_{ik} were observed, the log-likelihood function for the complete data is

$$\ell(\Theta) = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik}(\log\pi_k + \log f_k(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})). \quad (3)$$

For the purpose of variable selection, Pan and Shen (2006) proposed the regularized log-likelihood function

$$\ell_P(\Theta) = \ell(\Theta) - \lambda\sum_{k=1}^{K}\sum_{j=1}^{p}|\mu_{kj}|, \quad (4)$$

where the penalty function is the L_1-norm of the mean vectors, and we refer this model as the L_1-norm Gaussian mixture model (L_1-GMM). The L_1-norm penalty shrinks some of the fitted means μ_{kj} to be *exactly* zero when making λ sufficiently large. As we can see from (1), if for the jth variable, all the cluster-specific means μ_{kj}, $k = 1, \ldots, K$, are shrunken to zero, $\exp(-(x_j^* - \mu_{kj})^2/(2\sigma_j^2))$ becomes a common factor $\exp(-x_j^{*2}/(2\sigma_j^2))$ that does not depend on k; hence the jth variable does not contribute to the clustering score (1), and it can be removed.

In order to remove the jth variable, we need all corresponding μ_{kj}, $k = 1, \ldots, K$, to be zero. However, we can see from (4) that the L_1-norm penalty treats the μ_{kj} individually, that is, it does not use the information that μ_{kj} and $\mu_{k'j}$ are associated with the same variable x_j, and intuitively, they belong to one "group" and they should be treated differently from $\mu_{kj'}$, which is associated with a different variable $x_{j'}$. When the jth variable is noninformative, due to ignoring the "group" information in the data, the L_1-norm penalty tends to shrink only *some* of the μ_{kj}'s, but not *all* of them to be zero, hence it fails to exclude the jth variable. In the next two subsections, we propose two different penalty functions, that is, the L_∞-norm penalty and a hierarchical penalty, that incorporate the "group" information into the modeling procedure. When the jth variable is noninformative, comparing to the L_1-norm penalty, the two new penalties tend to shrink all μ_{kj}'s to be zero more effectively, hence they help improve the clustering performance. Our numerical results in Section 4 provide further supportive evidence.

2.1 *Model I: The Adaptive L_∞-norm Penalized Gaussian Mixture Model (ALP-GMM)*

For the ALP-GMM, we consider the penalized log-likelihood function:

$$\ell_P(\Theta) = \ell(\Theta) - \lambda\sum_{j=1}^{p}\max_k(|\mu_{1j}|, \ldots, |\mu_{kj}|, \ldots, |\mu_{Kj}|), \quad (5)$$

where $\max(|\mu_{1j}|, \ldots, |\mu_{Kj}|) = \|(\mu_{1j}, \ldots, \mu_{Kj})\|_\infty$. Different from penalizing every μ_{kj} individually, the L_∞-norm penalizes the maximum absolute value of μ_{kj}, $k = 1, \ldots, K$, for the jth variable. If the maximum of $|\mu_{kj}|$, $k = 1, \ldots, K$, is shrunken to zero, all μ_{kj} are automatically shrunken to zero. The L_∞-norm penalty has also been used in Zhang et al. (2006), Zhao et al. (2006), and Zou and Yuan (2006) for regression and classification problems.

To further improve the model (5), we apply the adaptive idea from Breiman (1995), Shen and Ye (2002), Zhang and Lu (2007), Zhao and Yu (2006), Zou (2006), and Yuan and Lin (2007), that is, to penalize different variables differently. We consider

$$\ell_P(\Theta) = \ell(\Theta) - \lambda\sum_{j=1}^{p}w_j \cdot \max_k(|\mu_{1j}|, \ldots, |\mu_{kj}|, \ldots, |\mu_{Kj}|), \quad (6)$$

where w_j are prespecified weights. The intuition is that if the jth variable is informative for clustering, we would like the corresponding w_j to be small, hence the jth variable is lightly penalized, whereas if the jth variable is noninformative for clustering, we would like the corresponding w_j to be large, hence the jth variable is heavily penalized. How to prespecify w_j from the data will be discussed in the numerical study section.

2.2 *Model II: The Adaptive Hierarchically Penalized Gaussian Mixture Model (AHP-GMM)*

The L_∞-norm penalty makes use of the information that μ_{kj} and $\mu_{k'j}$ are associated with the same variable by shrinking the maximum absolute value of μ_{kj} within the jth variable. If we denote $M_j = \max(|\mu_{1j}|, \ldots, |\mu_{Kj}|)$, the corresponding L_∞-norm penalty on the jth variable is λM_j, and we can write $\mu_{kj} = M_j\alpha_{kj}$, where $-1 \leq \alpha_{kj} \leq 1$. However, the L_∞-norm penalty tends to shrink the μ_{kj}, $k = 1, \ldots, K$ into the same magnitude, and the L_∞-norm penalty also tends to select μ_{kj}, $k = 1, \ldots, K$ in an "all-in-all-out" fashion. This motivates us to reparameterize μ_{kj} in a more general way:

$$\mu_{kj} = \gamma_j\theta_{kj}, \quad k = 1, \ldots, K; \quad j = 1, \ldots, p, \quad (7)$$

where $\gamma_j \geq 0$ (for identifiability reasons). Note that here γ_j plays a similar role as M_j, but it does not have to be the maximum of $|\mu_{kj}|$; similarly θ_{kj} does not have to be bounded between -1 and 1. This decomposition reflects the information that μ_{kj}, $k = 1, \ldots, K$, all belong to one variable x_j, by treating each μ_{kj} hierarchically. γ_j is at the first level of the hierarchy, controlling μ_{kj}, $k = 1, \ldots, K$, as a group; θ_{kj}'s are at the second level of the hierarchy, reflecting differences within the jth variable.

To estimate γ_j and θ_{kj}, we consider

$$\ell_P(\Theta) = \ell(\Theta) - \lambda_\gamma\sum_{j=1}^{p}\gamma_j - \lambda_\theta\sum_{k=1}^{K}\sum_{j=1}^{p}|\theta_{kj}|, \quad (8)$$

subject to $\gamma_j \geq 0$. Note that there are two tuning parameters, λ_γ and λ_θ. λ_γ controls the estimates at the variable-specific level, and it can effectively remove noninformative variables: if γ_j is shrunken to zero, all μ_{kj} for the jth variable will be equal to zero. λ_θ controls the estimates at the cluster-specific level: if γ_j is not equal to zero, some of the θ_{kj}, hence some of the μ_{kj}, $k = 1, \ldots, K$, still have the possibility of being zero; in this sense, the hierarchical penalty keeps the flexibility of the L_1-norm penalty.

The adaptive idea in (6) also applies here. If the jth variable is informative, we would like to penalize its γ_j and θ_{kj} lightly, and if the jth variable is noninformative, we would like to penalize its γ_j and θ_{kj} heavily. Hence we propose the AHP-GMM:

$$\ell_P(\Theta) = \ell(\Theta) - \lambda_\gamma\sum_{j=1}^{p}w_j^\gamma\gamma_j - \lambda_\theta\sum_{k=1}^{K}\sum_{j=1}^{p}w_{kj}^\theta|\theta_{kj}|, \quad (9)$$

where w_j^γ and w_{kj}^θ are prespecified weights.

3. Algorithms

In this section, we describe details of our algorithms for estimating the parameter Θ in ALP-GMM and AHP-GMM.

3.1 *The General EM Algorithm for Penalized GMM*

We consider

$$\ell_P(\Theta) = \ell(\Theta) - J(\Omega) \tag{10}$$

$$= \sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}(\log \pi_k + \log f_k(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})) - J(\Omega), \tag{11}$$

where $\Omega = \{\mu_{kj}, k=1,\ldots,K; j=1,\ldots,p\}$, $J(\Omega) = \lambda\sum_{j=1}^{p} w_j \times \max(|\mu_{1j}|,\ldots,|\mu_{Kj}|)$ for ALP-GMM, and $J(\Omega) = \lambda_\gamma \times \sum_{j=1}^{p} w_j^\gamma \gamma_j + \lambda_\theta \sum_{k=1}^{K}\sum_{j=1}^{p} w_{kj}^\theta |\theta_{kj}|$ for AHP-GMM. The indicators τ_{ik} are not observed, and the EM algorithm can be used to maximize the above penalized log likelihood with respect to Θ, and it follows closely to the EM algorithm for the standard nonpenalized GMM model (McLachlan and Peel, 2002). As we will see, the only difference exists in estimating μ_{kj} in the M-step.

The EM algorithm iterates between two alternating steps and produces a sequence of estimates $\hat{\Theta}^{(t)}, t = 0, 1, 2, \ldots$:

E-step: Impute values for unobserved τ_{ik} by

$$\begin{aligned}\hat{\tau}_{ik}^{(t)} &= \mathrm{E}\left[\tau_{ik} \,|\, \boldsymbol{x}_i, \hat{\Theta}^{(t)}\right] \\ &= \Pr\left[\tau_{ik} = 1 \,|\, \boldsymbol{x}_i, \hat{\Theta}^{(t)}\right] \\ &= \frac{\hat{\pi}_k^{(t)} f_k\left(\boldsymbol{x}_i; \hat{\boldsymbol{\mu}}_k^{(t)}, \hat{\boldsymbol{\Sigma}}^{(t)}\right)}{\sum_{k=1}^{K} \hat{\pi}_k^{(t)} f_k\left(\boldsymbol{x}_i; \hat{\boldsymbol{\mu}}_k^{(t)}, \hat{\boldsymbol{\Sigma}}^{(t)}\right)}, \\ & \qquad i = 1,\ldots,n; \quad k = 1,\ldots,K.\end{aligned}$$

Plug them into $\ell_P(\Theta)$ (10), yielding the so-called penalized Q-function:

$$\begin{aligned} Q_P&(\Theta, \hat{\Theta}^{(t)}) \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \hat{\tau}_{ik}(\log \pi_k + \log f_k(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})) - J(\Omega).\end{aligned}$$

M-step: Update the parameter estimates

$$\hat{\Theta}^{(t+1)} = \arg\max_{\Theta} Q_P(\Theta, \hat{\Theta}^{(t)}).$$

Specifically,

$$\frac{\partial Q_P}{\partial \pi_k} = 0 \Rightarrow \hat{\pi}_k^{(t+1)} = \sum_{i=1}^{n} \hat{\tau}_{ik}^{(t)} \big/ n, \quad k = 1,\ldots,K, \tag{12}$$

$$\begin{aligned}\frac{\partial Q_P}{\partial \sigma_j^2} &= 0 \Rightarrow \hat{\sigma}_j^{2,(t+1)} \\ &= \sum_{i=1}^{n}\sum_{k=1}^{K} \hat{\tau}_{ik}^{(t)} \left(x_{ij} - \hat{\mu}_{kj}^{(t)}\right)^2 \big/ n, \quad j = 1,\ldots,p,\end{aligned} \tag{13}$$

and

$$\begin{aligned}\hat{\Omega}^{(t+1)} = \arg\max_{\mu_{kj}} -\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{p} \\ \times \hat{\tau}_{ik}^{(t)} (x_{ij} - \mu_{kj})^2 \big/ \hat{\sigma}_j^{2,(t+1)} - J(\Omega).\end{aligned} \tag{14}$$

Notice that in $J(\Omega)$, both the L_∞-norm penalty and the hierarchical penalty have nondifferentiable points, so they pose optimization challenges, and we will consider (14) separately.

We also note that in the M-step, the penalized Q-function does not have explicit solutions. Its exact maximizer needs to be solved iteratively. Therefore, strictly speaking, our algorithm is an expectation-conditional maximization (ECM) algorithm (Meng and Rubin, 1993), which replaces the M-step of EM by a sequence of conditional maximization steps, each of which maximizes the penalized Q-function over Θ but with some elements of Θ fixed at their previous values. Using Theorem 3 in Meng and Rubin (1993), our algorithm is guaranteed to converge to a stationary point.

3.2 *Estimating μ_{kj} in ALP-GMM*

In ALP-GMM, (14) becomes

$$\begin{aligned}\min_{\mu_{kj}} \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\sum_{j=1}^{p} \tau_{ik}(x_{ij} - \mu_{kj})^2 \big/ \sigma_j^2 \\ + \lambda \sum_{j=1}^{p} w_j \cdot \max_k(|\mu_{1j}|,\ldots,|\mu_{Kj}|).\end{aligned}$$

This can be decomposed into p separate minimization problems

$$\begin{aligned}\min_{\mu_{kj}} \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}(x_{ij} - \mu_{kj})^2 \big/ \sigma_j^2 \\ + \lambda \cdot w_j \cdot \max_k(|\mu_{1j}|,\ldots,|\mu_{Kj}|), \quad 1 \le j \le p.\end{aligned} \tag{15}$$

For each j, (15) can be transformed into a quadratic programming problem:

$$\min_{\mu_{kj}, M_j} \quad \frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K} \tau_{ik}(x_{ij} - \mu_{kj})^2 \big/ \sigma_j^2 + \lambda \cdot w_j \cdot M_j, \tag{16}$$

$$\text{subject to} \quad -M_j \le \mu_{kj} \le M_j, \quad k = 1,\ldots,K, \tag{17}$$

$$M_j \ge 0. \tag{18}$$

Hence most commercially available packages can be used to solve it.

We have also explored explicit forms for the solutions to (15), which help us gain more insights into the nature of the L_∞-norm penalty. Let $\mu_{kj}^0 = \sum_{i=1}^{n} \tau_{ik} x_{ij} / \sum_{i=1}^{n} \tau_{ik}$, for $j = 1,\ldots,p$ and $k = 1,\ldots,K$, which are the solutions when there is no penalty (or $\lambda = 0$). We can show that $\hat{\mu}_{kj}$, the solution to the minimization problem (15), can be achieved by shrinking a weighted average of several μ_{kj}^0.

THEOREM 1. *For the jth minimization problem* (15), *if there exist* $k_1,\ldots,k_r$, *such that*

$$|\hat{\mu}_{k_1 j}| = \cdots = |\hat{\mu}_{k_r j}| > |\hat{\mu}_{kj}|, \quad \textit{for } k \notin \{k_1,\ldots,k_r\}, \tag{19}$$

then

$$\hat{\mu}_{kj} = \begin{cases} \mu^0_{kj} & k \notin \{k_1, \ldots, k_r\} \\ \mathrm{sgn}\left(\mu^0_{kj}\right) \left(\sum_{s=1}^{r} \frac{\tau_{\cdot k_s}}{\sum_{s=1}^{r} \tau_{\cdot k_s}} \left|\mu^0_{k_s j}\right| - \frac{\lambda w_j \sigma_j^2}{\sum_{s=1}^{r} \tau_{\cdot k_s}} \right)_+ & k \in \{k_1, \ldots, k_r\}, \end{cases} \tag{20}$$

where $\tau_{\cdot k_s} = \sum_{i=1}^n \tau_{ik_s}$; $(\cdot)_+$ *is the positive part of the argument.*

Details of the proof are in the Web Appendix 1. From Theorem 1, we can see when there are r maximums among $|\hat{\mu}_{kj}|$, only the corresponding μ^0_{kj} will be shrunken by the L_∞-norm penalty, and they are shrunken to the same absolute value. This value is based on a weighted average of μ^0_{kj} of the corresponding r clusters, and the weights are proportional to $\tau_{\cdot k}$. We can also see that if the jth variable is noninformative and all $|\mu^0_{kj}|$ are close to zero, then the L_∞-norm penalty tends to shrink all of them to zero (with an appropriately chosen λw_j).

To implement Theorem 1, we need to decide r, the number of maximums among $\hat{\mu}_{kj}$, and the set $\{k_1, \ldots, k_r\}$, which indicates which r μ^0_{kj} should be shrunken. When K is not very large, say $K \leq 10$, we can use an exhaustive search to find r and $\{k_1, \ldots, k_r\}$, that is, for each $1 \leq r \leq K$, we search over all possible sets $\{k_1, \ldots, k_r\}$. For each possible set, we estimate $\hat{\mu}_{kj}$ using (20), then check whether the estimate satisfies the assumption (19). If the assumption is satisfied, we compute the corresponding value for the objective function (15). Finally, we choose $\hat{\mu}_{kj}$ that gives the smallest value for the objective function. When K is large, we will resort to the quadratic programming (16)–(18).

3.3 *Estimating* μ_{kj} *in AHP-GMM*

In AHP-GMM, (14) becomes

$$\min_{\gamma_j, \theta_{kj}} \quad \frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{K} \sum_{j=1}^{p} \tau_{ik} (x_{ij} - \gamma_j \theta_{kj})^2 / \sigma_j^2 + \lambda_\gamma \sum_{j=1}^{p} w_j^\gamma \gamma_j + \lambda_\theta \sum_{k=1}^{K} \sum_{j=1}^{p} w^\theta_{kj} |\theta_{kj}| \tag{21}$$

$$\text{subject to} \quad \gamma_j \geq 0, \quad j = 1, \ldots, p. \tag{22}$$

We can use an iterative approach to estimate γ_j and θ_{kj}, that is, we first fix θ_{kj} and estimate γ_j, then we fix γ_j and estimate θ_{kj}, and we iterate between these two steps until the solution converges. Because at each step, the value of the objective function (21) decreases, the solution is guaranteed to converge. We have the following theorem that helps us solve for γ_j and θ_{kj} at each step.

Theorem 2. *Let* $\mu^0_{kj} = \sum_{i=1}^n \tau_{ik} x_{ij} / \sum_{i=1}^n \tau_{ik}$ *and* $\tau_{\cdot k} = \sum_{i=1}^n \tau_{ik}$.

- *When* θ_{kj}, $k = 1, \ldots, K$ *and* $j = 1, \ldots, p$, *are fixed,*

$$\hat{\gamma}_j = \mathbb{I}_{(\exists k, \theta_{kj} \neq 0)} \left(\sum_{k=1}^{K} \frac{\xi_k}{\sum_{k=1}^{K} \xi_k} \frac{\mu^0_{kj}}{\theta_{kj}} - \lambda_\gamma w_j^\gamma \frac{\sigma_j^2}{\sum_{k=1}^{K} \xi_k} \right)_+ , \tag{23}$$

where $\xi_k = \tau_{\cdot k} \theta^2_{kj}$.

- *When* γ_j, $j = 1, \ldots, p$, *are fixed,*

$$\hat{\theta}_{kj} = \mathbb{I}_{(\gamma_j > 0)} \cdot \mathrm{sgn}\left(\mu^0_{kj}\right) \left(\frac{|\mu^0_{kj}|}{\gamma_j} - \frac{\lambda_\theta w^\theta_{kj}}{\gamma_j^2} \frac{\sigma_j^2}{\tau_{\cdot k}} \right)_+ . \tag{24}$$

Equations (23) and (24) show that both $\hat{\gamma}_j$ and $\hat{\theta}_{kj}$ are soft-thresholding estimates. Details of the proof are in the Web Appendix 2. Here we give some intuitive explanation.

We first look at $\hat{\gamma}_j$ (23). If all θ_{kj} are zero, it is natural to estimate γ_j also to be zero because of the penalty on γ_j. If not all θ_{kj} are 0, say, $\theta_{k_1 j}, \ldots, \theta_{k_r j}$ are not zero, then from our reparameterization, we have $\gamma_j = \mu_{k_s j} / \theta_{k_s j}, 1 \leq s \leq r$. Plugging in $\mu^0_{k_s j}$ for $\mu_{k_s j}$, we obtain r estimates for $\gamma_j : \tilde{\gamma}_j = \mu^0_{k_s j} / \theta_{k_s j}, 1 \leq s \leq r$. A natural estimate for γ_j is then a weighted average of the $\tilde{\gamma}_j$, and equation (23) provides such a (shrunken) average, with weights proportional to ξ_k.

Now consider $\hat{\theta}_{kj}$ (24). If $\gamma = 0$, it is natural to estimate all θ_{kj} to be also zero because of the penalty on θ_{kj}. When $\gamma_j > 0$, we have $\theta_{kj} = \mu_{kj} / \gamma_j$. Again, plugging in μ^0_{kj} for μ_{kj}, we obtain $\tilde{\gamma} = \mu^0_{kj} / \gamma_j$. Equation (24) shrinks $\tilde{\gamma}$ and the amount of shrinkage is inversely proportional to γ_j^2. So when γ_j is large, which indicates the jth variable is informative, the amount of shrinkage is small, whereas when γ_j is small, which indicates the jth variable is less informative, the amount of shrinkage is large.

4. Simulation Study

In this section, we use simulation data to demonstrate our methods ALP-GMM and AHP-GMM, and compare the results with those of the L_1-GMM (Pan and Shen, 2006), COSA (Friedman and Meulman, 2004), and MBSC (Hoff, 2006).

We first mimicked the simulation in Pan and Shen (2006). There were a total of $p = 1000$ variables with the first 150 informative, whereas the other 850 noninformative in forming two clusters. Specifically, the first 150 variables were independent and identically distributed (i.i.d.). $N(0, 1)$ for the first cluster and i.i.d. $N(1.5, 1)$ for the second cluster, whereas the remaining 850 variables were all i.i.d. $N(0, 1)$ for both clusters.

We generated $n = 100$ observations, with 85 in the first cluster and 15 in the second one. We denote this setting as "85-15." We computed the weights w_j in (6), w_j^γ and w^θ_{kj} in (9) using the unpenalized estimates $\mu^0_{kj} = \sum_{i=1}^n \tau_{ik} x_{ij} / \sum_{i=1}^n \tau_{ik}$. Specifically

$$\begin{aligned} M^0_j &= \max_k \left(\left|\mu^0_{1j}\right|, \ldots, \left|\mu^0_{Kj}\right|\right), \\ w_j &= 1/M^0_j, \\ w_j^\gamma &= 1/M^0_j, \\ w^\theta_{kj} &= 1/\left|\mu^0_{kj}\right|. \end{aligned}$$

Table 1

Simulation results for the "85-15" example. "$K = 2$" is the number of times (out of 50) that 2 was identified as the number of clusters. "Error rate" is the average proportion of wrongly clustered data points. "Info" is the average number of selected informative variables (out of 150). "Noninfo" is the average number of noninformative variables (out of 850) that were kept. The numbers in the parentheses are the corresponding standard deviations. "GMM without noise" is to apply the standard GMM method on the data set with only the first 150 informative variables, and its "Error rate" can be considered as a benchmark. "GMM" is the standard GMM method using all 1000 variables.

Method	$K = 2$	Error rate	Info	Noninfo
GMM without noise	50	0 (0)	—	—
GMM	0	—	—	—
L_1-GMM	50	0 (0)	149.2 (1.2)	17.9 (6.0)
ALP-GMM	50	0 (0)	148.0 (1.9)	2.1 (1.8)
AHP-GMM	50	0 (0)	148.5 (1.5)	5.7 (2.5)
MBSC	50	0 (0)	148.7 (1.3)	96.2 (16.3)
COSA	—	0.02 (0.01)	—	—

We chose the tuning parameters and the number of clusters using the Bayesian Information Criterion (BIC; Schwarz, 1978):

$$\mathrm{BIC} = -2\sum_{i=1}^{n}\log\left(\sum_{k=1}^{K}\hat{\pi}_k f_k(\boldsymbol{x}_i;\hat{\boldsymbol{\mu}}_k,\hat{\boldsymbol{\Sigma}})\right) + P\log n, \qquad (25)$$

where P is the total number of nonzero estimates in $\hat{\mu}_{kj}, \hat{\sigma}_j^2$, and $\hat{\pi}_k$.

We repeated this 50 times, recorded the number of times that the true number of clusters was detected, and computed the average misclustering error rates, and their corresponding standard deviations. For ALP-GMM, AHP-GMM, L_1-GMM, and MBSC, we also recorded the number of selected informative variables and the number of noninformative variables that were kept. The results are summarized in Table 1. Note that COSA does not explicitly select variables but rather assigns variables with different importance scores. Figure 2 shows one typical plot of these importance scores.

As we can see, the three penalized GMM methods always selected $K = 2$ as the number of clusters for the 50 repetitions. In contrast, the GMM method without any penalty always selected $K = 1$, and was not able to discover the "true" two-cluster data structure. This result is not surprising: Based on the first 150 variables, there were two clusters, but if based on the other 850 variables, there was indeed only one cluster; the first 150 variables were overwhelmed by the other 850 variables when using all 1,000 variables. We can also see that our ALP-GMM and AHP-GMM methods performed similarly to the L_1-GMM method in terms of selecting informative variables, but tended to keep fewer noninformative variables. The MBSC method also detected the true two-cluster data structure and had zero misclustering error, but it selected more noninformative variables than ALP-GMM and AHP-GMM. For COSA, we forced the number of clusters to be equal to the truth, that is, two. The misclustering error for COSA was small; however, the importance scores from COSA (Figure 2) did not seem to set a clear distinction between the informative variables and the noninformative variables.

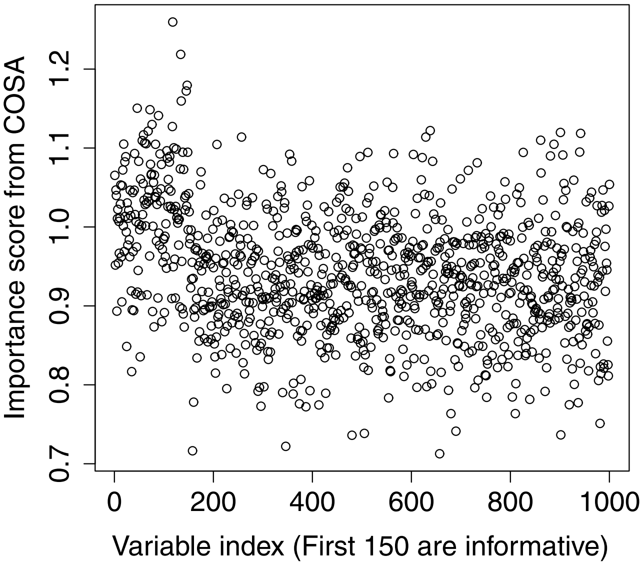

Figure 2. Importance scores from COSA for the "85-15" example.

We then considered two three-cluster scenarios. In both scenarios, there were a total of $p = 402$ variables with the first 2 informative and the other 400 noninformative in forming three clusters. The first 2 variables were i.i.d. $N(0, 1)$ for the first cluster, i.i.d. $N(2.5, 1)$ for the second cluster, and i.i.d. $N(5, 1)$ for the third cluster, whereas the remaining 400 variables were all i.i.d. $N(0, 1)$ for all three clusters. In the first scenario, we generated 20 observations for each of the first cluster and the third cluster, and 100 for the second cluster. We denote it as "20-100-20." In the second scenario, we generated 50 observations for each of the first cluster and the third cluster, and 20 for the second cluster. We denote it as "50-20-50." Similar to the previous simulation, we repeated this 50 times, recorded the number of times that the true number of clusters was detected, and computed the average misclustering error rates, the average number of selected informative variables, the average number of noninformative variables that were kept, and their corresponding standard deviations. The results are summarized in Table 2. We note that the numbers in the table are based on the repetitions where the true number of clusters was detected. For COSA, we enforced the number of clusters to be equal to the truth, that is, three.

As we can see, our ALP-GMM and AHP-GMM methods discovered the three-cluster data structure for almost every repetition (out of 50), and the clustering error rates were just slightly higher than that of the GMM method without using any of the noninformative variables, that is, the "oracle." The ALP-GMM method removed all 400 noninformative variables for every repetition, and the AHP-GMM method removed all 400 noninformative variables for almost every repetition.

Table 2

Simulation results for the "20-100-20" example and the "50-20-50" example: the upper part is for the "20-100-20" example, and the lower part is for the "50-20-50" example. Descriptions for the columns are the same as those in the caption of Table 1.

Method	$K = 3$	Error rate	Info	Noninfo
The "20-100-20" example				
GMM without noise	49	0.049 (0.020)	—	—
GMM	0	—	—	—
L_1-GMM	0	—	—	—
ALP-GMM	48	0.051 (0.021)	2 (0)	0 (0)
AHP-GMM	48	0.051 (0.022)	2 (0)	0.13 (0.44)
MBSC	7	0.35 (0.08)	0.82 (0.85)	34.28 (39.01)
COSA	—	0.44 (0.10)	—	—
The "50-20-50" example				
GMM without noise	48	0.045 (0.020)	—	—
GMM	0	—	—	—
L_1-GMM	6	0.050 (0.021)	2 (0)	2.5 (1.52)
ALP-GMM	48	0.050 (0.021)	2 (0)	0.02 (0.14)
AHP-GMM	48	0.048 (0.020)	2 (0)	0.21 (0.58)
MBSC	6	0.48 (0.21)	0.26 (0.53)	35.60 (46.45)
COSA	—	0.54 (0.04)	—	—

The L_1-GMM method failed to detect the true three-cluster structure in all 50 repetitions for the "20-100-20" example, and in most of the repetitions for the "50-20-50" example. The MBSC method failed to detect the true three-cluster structure in most of the repetitions for both the "20-100-20" example and the "50-20-50" example.

5. Real Data Analysis

In this section, we apply the ALP-GMM and the AHP-GMM methods to two gene microarray data sets.

The first data set we considered is the Leukemia Dataset in Golub et al. (1999). This data set consists of 38 training data and 34 test data for two types of acute leukemia—acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). Each sample is a vector of $p = 7129$ genes. We applied the ALP-GMM and the AHP-GMM methods to the training data, ignoring the class labels. The tuning parameters and the number of clusters were chosen using BIC, and the chosen model was evaluated on the test data. The results are summarized in Table 3. As we can see, both the ALP-GMM and the AHP-GMM selected $K = 2$ as the number of clusters, which agrees with the true number of classes. The ALP-GMM selected 20 genes and had 2 misclustering errors on the training data; the AHP-GMM selected 25 genes and had 1 "misclustering error" on the training data. Based on the genes selected from the training data, the ALP-GMM had 3 misclustering errors on the test data, and the AHP-GMM had two. The 20 genes selected by ALP-GMM is a subset of the 25 genes selected by AHP-GMM.

Table 3

Results on the real data sets: the upper part is for the Leukemia Dataset, and the lower part is for the SRBCT Dataset

Method	K	Genes	Training error	Testing error
The Leukemia Dataset				
Golub et al. (1999)	—	50	3/38	4/34
ALP-GMM	2	20	2/38	3/34
AHP-GMM	2	25	1/38	2/34
The SRBCT Dataset				
Kahn et al. (2001)	—	96	0/63	0/20
ALP-GMM	4	44	0/63	0/20
AHP-GMM	4	49	0/63	0/20

The second data set we considered consists of microarray experiments of small round blue cell tumors (SRBCT) of childhood cancer (Khan et al., 2001). The tumors are classified as Burkitt lymphoma (BL), Ewing sarcoma (EWS), neuroblastoma (NB), or rhabdomyosarcoma (RMS). A total of 63 training samples and 20 test samples were provided. Each sample consists of expression measurements on $p = 2308$ genes. We analyzed this data set in the similar way as with the Leukemia Dataset. The results are summarized in Table 3. Both the ALP-GMM and the AHP-GMM selected $K = 4$ as the number of clusters. The training errors and the test errors are all zero. The ALP-GMM selected 44 genes, and the AHP-GMM selected 49 genes. The 44 genes selected by ALP-GMM and the 49 genes selected by AHP-GMM have 28 overlapping genes. Due to lack of space, only the heatmap for the 49 genes selected by AHP-GMM is shown (Figure 3). Clear separation of the four clusters is evident.

6. Conclusion

In this article, we have proposed two methods, ALP-GMM and AHP-GMM, for simultaneously clustering high-dimensional data and selecting informative variables. Our methods are in the framework of penalized model-based clustering. In particular, we penalize $\hat{\mu}_{kj}$'s, the differences between the cluster means and the overall mean for variable x_j. If all $\hat{\mu}_{kj}$'s, $k = 1, \ldots, K$, are shrunken to zero, we consider x_j as noninformative. Unlike the L_1-norm penalization, the penalty terms that we consider make use of the fact that

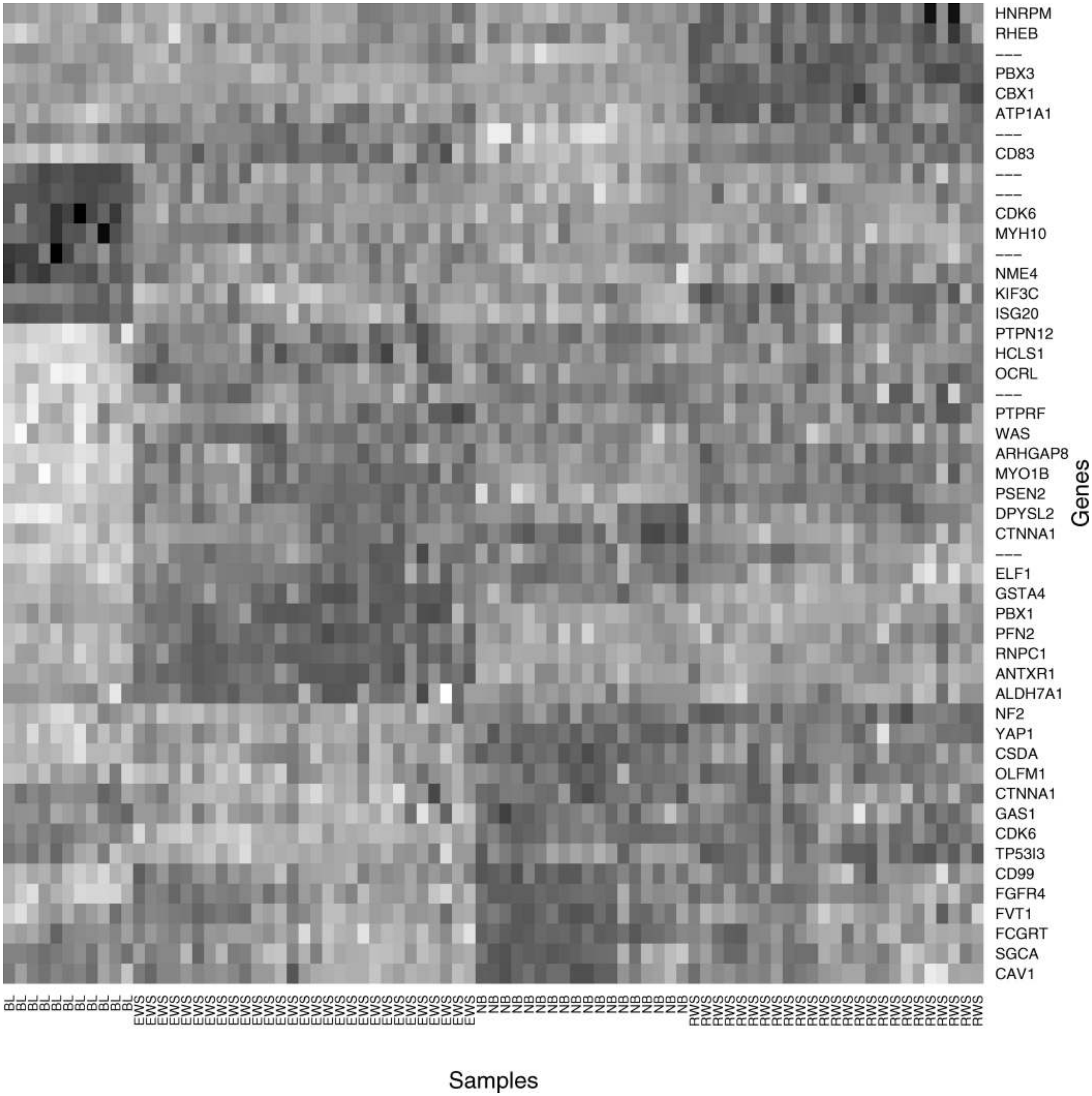

Figure 3. Heatmap of the 49 genes selected by AHP-GMM from the SRBCT Dataset.

parameters belonging to one variable can be treated as a natural "group." We have presented some evidence that the two new methods tend to remove noninformative variables more effectively and provide better clustering results than the L_1-norm approach.

In terms of the clustering error rate and variable selection, the two new methods performed similarly in our simulations. However, in terms of the computational cost, because ALP-GMM relies on either a quadratic programming or on an exhaustive search, whereas AHP-GMM has closed-form solutions at each step, we have found that AHP-GMM tends to run faster than ALP-GMM.

7. Supplementary Materials

Web Appendix 1 in Section 3.2 and Web Appendix 2 in Section 3.3 are available under the Paper Information link at the *Biometrics* website `http://www.biometrics.tibs.org`.

Acknowledgements

We thank the reviewers, the associate editor, and the editor N. Wang for their thoughtful and helpful comments. Wang and Zhu are partially supported by grant DMS-0505432 from the National Science Foundation.

References

Bickel, P. J. and Levina, L. (2004). Some theory for fisher's linear discriminant function, "naive Bayes," and some alternatives when there are many more variables than observations. *Bernoulli* **10,** 989–1010.

Breiman, L. (1995). Better subset regression using the nonnegative garrote. *Technometrics* **37,** 373–384.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society, Series B* **39,** 1–38.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96,** 1348–1360.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association* **97,** 611–631.

Friedman, J. H. and Meulman, J. J. (2004). Clustering objects on subsets of attributes (with discussion). *Journal of the Royal Statistical Society, Series B* **66,** 815–849.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., and Bloomfield, C. D. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science* **286,** 531–537.

Hoff, P. D. (2006). Model-based subspace clustering. *Bayesian Analysis* **1,** 321–344.

Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., and Meltzer, P. S. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature Medicine* **7,** 673–679.

Liu, J. S., Zhang, J. L., and Palumbo, M. J. (2003). Bayesian clustering with variable and transformation selection (with discussion). *Bayesian Statistics* **7,** 249–275.

Marron, J. and Todd, M. (2002). *Distance weighted discrimination.* Technical Report, School of Operations Research and Industrial Engineering, Cornell University, Ithaca, NY.

McLachlan, G. and Peel, D. (2002). *Finite Mixture Models.* New York: John Wiley & Sons.

Meng, X. L. and Rubin, D. (1993). Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika* **80,** 267–278.

Pan, W. and Shen, X. (2006). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research* **8,** 1145–1164.

Raftery, A. E. (2003). Discussion of "Bayesian clustering with variable and transformation selection" by Liu et al. *Bayesian Statistics* **7,** 266–271.

Raftery, A. E. and Dean, N. (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association* **101,** 168–178.

Schwarz, G. (1978). Estimating the dimension of a model. *Annals of Statistics* **6,** 461–464.

Shen, X. and Ye, J. (2002). Adaptive model selection. *Journal of the American Statistical Association* **97,** 210–221.

Tadesse, M. G., Sha, N., and Vannucci, M. (2005). Bayesian variable selection in clustering high-dimensional data. *Journal of the American Statistical Association* **100,** 602–617.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society, Series B* **58,** 267–288.

Yuan, M. and Lin, Y. (2007). On the nonnegative garrote estimator. *Journal of the Royal Statistical Society, Series B* **69,** 143–161.

Zhang, H. H. and Lu, W. (2007). Adaptive-LASSO for Cox's proportional hazard model. *Biometrika* **94,** 691–703.

Zhang, H. H., Liu, Y., Wu, Y., and Zhu, J. (2006). *Variable selection for multicategory SVM via sup-norm regularization.* Institute of Statistics Mimeo Series 2596, North Carolina State University, Raleigh, NC.

Zhao, P. and Yu, B. (2006). On model selection consistency of Lasso. *Journal of Machine Learning Research* **7,** 2541–2567.

Zhao, P., Rocha, G., and Yu, B. (2006). *Grouped and hierarchical model selection through composite absolute penalties.* Technical Report 703, Department of Statistics, University of California at Berkeley.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association* **101,** 1418–1429.

Zou, H. and Yuan, M. (2006). The F_∞-norm support vector machine. *Statistica Sinica*, in press.

Received November 2006. *Revised May* 2007.
Accepted July 2007.