Published in final edited form as:
Biometrics. 2010 September ; 66(3): 793–804. doi:10.1111/j.1541-0420.2009.01341.x.

Pairwise Variable Selection for High-dimensional Model-based Clustering

Jian Guo, **Elizaveta Levina**, **George Michailidis**, and **Ji Zhu**
Department of Statistics, University of Michigan, Ann Arbor, MI 48109, U.S.A.

Jian Guo: guojian@umich.edu; Elizaveta Levina: elevina@umich.edu; George Michailidis: gmichail@umich.edu; Ji Zhu: jizhu@umich.edu

SUMMARY

Variable selection for clustering is an important and challenging problem in high-dimensional data analysis. Existing variable selection methods for model-based clustering select informative variables in a "one-in-all-out" manner; that is, a variable is selected if at least one pair of clusters is separable by this variable and removed if it cannot separate any of the clusters. In many applications, however, it is of interest to further establish exactly which clusters are separable by each informative variable. To address this question, we propose a pairwise variable selection method for high-dimensional model-based clustering. The method is based on a new pairwise penalty. Results on simulated and real data show that the new method performs better than alternative approaches which use ℓ_1 and ℓ_∞ penalties and offers better interpretation.

Keywords

EM algorithm; Gaussian mixture models; Model-based clustering; Pairwise fusion; Variable selection

1. Introduction

The goal of clustering is to organize data into a small number of homogeneous groups, thus aiding interpretation. Clustering techniques have been employed in a wide range of scientific-fields, including biology, physics, chemistry and psychology. These techniques can broadly be classified into two categories: hierarchical methods and partition methods (see Gordon (2008), Kaufman and Rousseeuw (1990), and references therein). The former typically start from a dissimilarity matrix that captures differences between the objects to be clustered and produce a family of cluster solutions, whose main property is that any two clusters in the family are either disjoint or one is a superset of the other. Various popular agglomerative algorithms, such as single, complete and average linkage belong to this class. Partition algorithms produce non-overlapping clusters, whose defining characteristic is that distances between objects belonging to the same cluster are in some sense smaller than distances between objects in different clusters. The popular K-means algorithm (MacQueen, 1967) and its variants are members of this class. A statistically motivated partition method is model-based clustering, which models the data as a sample from a Gaussian mixture distribution, with each component corresponding to a cluster (McLachlan and Basford, 1988). A number of extensions addressing various aspects of this approach have recently appeared in the literature. For example, Banfield and Raftery (1993) generalized model-

Correspondence to: Ji Zhu, jizhu@umich.edu.

based clustering to the non-Gaussian case, while Fraley (1993) extended it to incorporate hierarchical clustering techniques.

The issue of variable selection in clustering, also known as subspace clustering, has started receiving increased attention in the literature recently (for a review of some early algorithms see Parsons et al. (2004)). For example, Friedman and Meulman (2004) proposed a hierarchical clustering method which uncovers cluster structure on separate subsets of variables; Tadesse et al. (2005) formulated the clustering problem in Bayesian terms and developed an MCMC sampler that searches for models comprised of different clusters and subsets of variables; Hoff (2006) also employed a Bayesian formulation based on a Polya urn model; and Raftery and Dean (2006) introduced a method to sequentially compare two nested models to determine whether a subset of variables should be included or excluded from the current model. Some recent approaches addressing variable selection are based on a regularization framework. Specifically, Pan and Shen (2006) proposed to maximize the Gaussian mixture likelihood while imposing an ℓ_1 penalty on the cluster means. In addition, the means of all clusters were required to sum up to zero for each variable. This method removes variables for which all cluster means are shrunk to zero and hence regarded as uninformative. Wang and Zhu (2007) treated the cluster mean parameters associated with the same variable as a natural "group" and proposed an adaptive ℓ_∞ penalty and an adaptive hierarchical penalty to make use of the available group information. Finally, Jornsten and Keles (2008) introduced mixture models that lead to sparse cluster representations in complex multifactor experiments.

All the existing variable selection methods for model-based clustering choose informative variables in a "one-in-all-out" manner; that is, a variable is selected if it is informative for at least one pair of clusters and removed only if it is non-informative for all clusters. However, in many practical situations, one may be interested in identifying which variables are discriminative for which specific pairs of clusters. A toy example illustration of such a scenario is shown in Figure 1. There are three clusters present in this two-dimensional data set; the first variable discriminates between clusters 2 and 3, while the second variable discriminates between clusters 1 and 2. We believe that such situations arise often in high-dimensional data, for example, in data obtained from high-throughput expression technologies.

To address this problem, this paper proposes a *pairwise* variable selection method for high-dimensional model-based clustering. Specifically, a *pairwise fusion* penalty is introduced to penalize the difference between (all) pairs of cluster centers for each variable and shrink the centroids of non-separable clusters to some identical value. If all cluster centroids associated with a variable are "fused," this variable is regarded as non-informative and removed from the model. Otherwise, the pairwise fusion penalty has the exibility of only fusing the centroids of non-separable clusters for this variable.

The remainder of the paper is organized as follows: Section 2 introduces the pairwise fusion penalty, and Section 3 discusses algorithmic issues. The performance of the proposed clustering technique on synthetic and real data is demonstrated in Sections 4 and 5, respectively. Finally, some concluding remarks are drawn in Section 6.

2. Problem Formulation and Pairwise Fusion

Suppose n samples have been collected on p variables and organized in a data matrix $\boldsymbol{X} = (x_{i,j})_{n\times p}$. Without loss of generality we can assume that the data are centered for each variable, i.e., $\sum_{i=1}^{n} x_{i,j}=0$, for all $1 \le j \le p$. In model-based clustering, a K-cluster problem is

described by a K-component Gaussian mixture model. Specifically, the observations $\boldsymbol{x}_i = (x_{i,1}, \ldots, x_{i,p})$ are assumed to be independent and generated from the density

$$f(\boldsymbol{x}_i) = \sum_{k=1}^{K} w_k \phi(\boldsymbol{x}_i; \mu_k, \sum_k), \tag{1}$$

where $\phi(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \sum_k)$ denotes the Gaussian density function with mean vector $\boldsymbol{\mu}_k = (\mu_{k,1}, \ldots, \mu_{k,p})$ and covariance matrix $\sum_k$,

$$\phi(\boldsymbol{x}_i; \mu_k, \sum_k) = \frac{1}{(2\pi)^{p/2} \det(\sum_k)^{1/2}} \exp\{-\frac{1}{2}(\boldsymbol{x}_i - \mu_k) \sum_k^{-1} (\boldsymbol{x}_i - \mu_k)^{\mathrm{T}}\}. \tag{2}$$

The "weights" w_k's ($w_k \geq 0$ for all $1 \leq k \leq K$ and $\sum_{k=1}^{K} w_k = 1$) are the mixing coefficients, capturing the contribution of the k-th cluster. We also introduce the following notation: the mean parameters $\mu_{k,j}$'s can be collected in a $K \times p$ matrix, with rows corresponding to clusters and columns to variables,

$$\mu = \begin{bmatrix} \mu_{1,1} & \mu_{1,2} & \cdots & \mu_{1,j} & \cdots & \mu_{1,p} \\ \mu_{2,1} & \mu_{2,2} & \cdots & \mu_{2,j} & \cdots & \mu_{2,p} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mu_{K,1} & \mu_{K,2} & \cdots & \mu_{K,j} & \cdots & \mu_{K,p} \end{bmatrix}.$$

We use $\boldsymbol{\mu}_k = (\mu_{k,1}, \ldots, \mu_{k,p})$ to represent the mean parameters for the k-th cluster (k-th row vector of $\boldsymbol{\mu}$), and $\boldsymbol{\mu}_{(j)} = (\mu_{1,j}, \ldots, \mu_{K,j})^{\mathrm{T}}$ to represent the mean parameters for the j-th variable (j-th column vector of $\boldsymbol{\mu}$).

The log-likelihood of the data matrix $\boldsymbol{X}$ is then given by,

$$\log p(\boldsymbol{X}|\boldsymbol{\Theta}) = \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} w_k \phi(\boldsymbol{x}_i; \mu_k, \sum_k) \right\}, \tag{3}$$

where $\boldsymbol{\Theta} = \{w_k, \mu_k, \sum_k\}_{k=1}^{K}$ is the parameter set of interest. The log-likelihood (3) can be maximized using an expectation-maximization (EM) algorithm, which in the E-step imputes the cluster membership of the samples and in the M-step estimates the mixing coefficients, the mean parameters and the covariance matrices. The number of clusters K can be selected using, for example, a Bayesian information criterion (BIC) or another similar criterion.

Given the estimate $\hat{\boldsymbol{\Theta}}$, an observation $\boldsymbol{x}^* = (x_1^*, \ldots, x_p^*)$ is assigned to the cluster which achieves

$$\arg \max_{1 \leq k \leq K} \widehat{w}_k \phi(\boldsymbol{x}^*; \widehat{\mu}_k, \widehat{\sum_k}). \tag{4}$$

2.1 The Pairwise Fusion Penalty

Since our focus here is on variables defined as informative in terms of differences in the cluster *means*, we make a further simplifying assumption that the covariance matrices are

the same for all clusters and are diagonal, i.e., $\sum_k = \sum = \text{diag}(\sigma_1^2, \sigma_2^2, \ldots, \sigma_p^2)$ for all $1 \le k \le K$. An alternative would be to impose a shrinkage penalty on the covariance matrices as well as the means, as in Xie et al. (2008), and consider a variable non-informative for a pair of clusters only if it has both the same mean and the same covariance structure in both clusters. This does not seem to be important for the applications we have in mind, such as gene selection in expression data clustering, since the main effects are normally contained in the means. Moreover, this is a common assumption in high-dimensional settings, since it significantly reduces the number of parameters to be estimated. There is also theoretical justification for estimating the covariance matrix by a diagonal matrix for discriminant analysis in high dimensions (Bickel and Levina, 2004). In addition, imposing an additional penalty on the variances results in a dramatic increase in computational cost, and, in our experience, very small empirical gains.

Given our focus on pairwise variable selection, we propose maximizing the following criterion for estimating the parameters of the Gaussian mixture model:

$$\sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} w_k \phi(\boldsymbol{x}_i; \mu_k, \sum) \right\} - \lambda \sum_{j=1}^{p} \sum_{1 \le k \le k' \le K} |\mu_{k,j} - \mu_{k',j}|, \tag{5}$$

where λ is a tuning parameter. We refer to $\sum_{j=1}^{p} \sum_{1 \le k < k' \le K} |\mu_{k,j} - \mu_{k',j}|$ as the *pairwise fusion penalty* (PFP). The aim of the penalty is to shrink the difference between every pair of cluster centers for each variable *j*. Due to the singularity of the absolute value function, some differences are shrunken to exactly zero, resulting in some cluster means $\hat{\mu}_{k,j}$'s having identical values. Notice that we are not shrinking the means to zero, only towards each other; zero has no special meaning here and the data do not need to be centered. If $\hat{\mu}_{k,j} = \hat{\mu}_{k',j}$, then variable *j* is considered to be "non-informative" for separating cluster *k* and cluster k', though it may be informative for separating other clusters. Moreover, if all cluster means for a variable are shrunken to the same value, that variable is considered non-informative for clustering purposes and can be removed from the model.

2.2 The Adaptive Pairwise Fusion Penalty

To further improve on (5), we apply the popular adaptive penalization (Zou, 2006) by considering

$$\sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} w_k \phi(\boldsymbol{x}_i; \mu_k, \sum) \right\} - \lambda \sum_{j=1}^{p} \sum_{1 \le k < k' \le K} \tau_{k,k'}^{(j)} |\mu_{k,j} - \mu_{k',j}|, \tag{6}$$

where $\tau_{k,k'}^{(j)}$ are pre-specified weights. We call this version adaptive pairwise fusion penalty (APFP). The intuition is that if variable *j* is informative for separating clusters *k* and k', we would like the corresponding $\tau_{k,k'}^{(j)}$ to be small; thus, the difference between $\mu_{k,j}$ and $\mu_{k',j}$ is lightly penalized. On the other hand, for a non-informative variable *j* for clusters *k* and k', we would like the corresponding $\tau_{k,k'}^{(j)}$ to be large and hence the difference between $\mu_{k,j}$ and $\mu_{k',j}$ is heavily penalized. In our implementation, we compute the weights from the unpenalized estimates as

$$\tau_{k,k'}^{(j)} = \left|\tilde{\mu}_{k,j} - \tilde{\mu}_{k',j}\right|^{-1},$$

where $\tilde{\mu}_{k,j}$ is the estimate of $\mu_{k,j}$ without any penalization ($\lambda = 0$).

It is interesting to compare our approach to the ℓ_1-regularized method proposed by Pan and Shen (2006) and the ℓ_∞-regularized method proposed by Wang and Zhu (2007). Note that Pan and Shen (2006) proposed an ℓ_1 penalty without adaptive weights, but for a fair comparison here we use adaptive versions of all the methods. Pan and Shen (2006) proposed to use the criterion,

$$\sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} w_k \phi(\boldsymbol{x}_i; \mu_k, \sum) \right\} - \lambda \sum_{j=1}^{p} \sum_{k=1}^{K} \tau_{k,j}^{\ell_1} \left|\mu_{k,j}\right|, \tag{7}$$

where $\tilde{\tau}_{k,j}^{\ell_1}$'s are adaptive weights defined as $\tau_{k,j}^{\ell_1} = 1/\left|\tilde{\mu}_{k,j}\right|$ for all $1 \le k \le K$ and $1 \le j \le p$. Here $\tilde{\mu}_{k,j}$ is the estimate from model-based clustering method without penalty. Notice that the data are required to be centered, and the ℓ_1 penalty shrinks the individual $\mu_{k,j}$'s towards zero (the global mean) and removes variable j from the model if all $\hat{\mu}_{k,j}$ for $1 \le k \le K$ are set to zero. However, it cannot identify variables that are non-informative for separating particular subsets of clusters, especially when the common mean of these clusters is different from zero. On the other hand, the ℓ_∞-regularized criterion proposed by Wang and Zhu (2007) is

$$\sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} w_k \phi(\boldsymbol{x}_i; \mu_k, \sum) \right\} - \lambda \sum_{j=1}^{p} \tau_j^{\ell_\infty} \max\left(\left|\mu_{1,j}\right|, \ldots, \left|\mu_{k,j}\right|, \ldots, \left|\mu_{K,j}\right|\right), \tag{8}$$

where the adaptive weight $\tau_j^{\ell_\infty} = 1/\max\left(\left|\tilde{\mu}_{1,j}\right|, \ldots, \left|\tilde{\mu}_{k,j}\right|, \ldots, \left|\tilde{\mu}_{K,j}\right|\right)$. Unlike the ℓ_1 penalty which shrinks each $\mu_{k,j}$ individually, the ℓ_∞ norm penalizes the maximum magnitude of the cluster means for each variable. If the largest cluster mean for variable j is shrunk to zero, then all other means for the j-th variable are automatically zero, and the variable can be eliminated from the model. However, this penalty is also unable to identify specific clusters that can be separated by a particular variable.

2.3 Model Selection

There are two parameters to be selected, the number of clusters K and the tuning parameter λ. We select them using a BIC-type criterion, defined by

$$\mathrm{BIC}(K, \lambda) = -2 \sum_{i=1}^{n} \log \left\{ \sum_{k=1}^{K} \widehat{w}_k \phi(\boldsymbol{x}_i; \widehat{\mu}_k, \widehat{\sum}) \right\} + d \log n, \tag{9}$$

where $\{\widehat{w}_k, \widehat{\mu}_k, \widehat{\sum}\}_{k=1}^{K}$ are estimated with K clusters and the tuning parameter λ. The degrees of freedom d are defined as the number of distinct nonzero estimates. Specifically, $d = K - 1 + p + e(\hat{\boldsymbol{\mu}})$, where $e(\hat{\boldsymbol{\mu}})$ is the number of distinct nonzero elements in $\{\hat{\mu}_{k,j}\}$. This definition is similar to the degrees of freedom for fused Lasso (Tibshirani et al., 2005).

3. The Optimization Algorithm

The optimization of the objective function (6) is non-trivial. As in classical model-based clustering, we employ an EM algorithm to maximize the log-likelihood function subject to the penalty constraint. Let $\Delta_{i,k}$ be the indicator of whether $\boldsymbol{x}_i$ is from cluster k, that is, $\Delta_{i,k} = 1$ if $\boldsymbol{x}_i$ belongs to cluster k, and $\Delta_{i,k} = 0$ otherwise. If the missing data $\Delta_{i,k}$ were observed, the penalized log-likelihood function for the complete data is given by

$$\sum_{i=1}^{n}\sum_{k=1}^{K}\Delta_{i,k}\{\log w_k+\log \phi(\boldsymbol{x}_i;\mu_k, \sum)\} - \lambda\sum_{j=1}^{p}\sum_{1\le k<k'\le K}\tau_{k,k'}^{(j)}\left|\mu_{k,j}-\mu_{k',j}\right|. \tag{10}$$

Our algorithm follows closely the EM algorithm for the standard (unpenalized) Gaussian mixture model (McLachlan and Peel, 2002); the main difference is in estimating $\mu_{k,j}$ in the M-step. The EM algorithm iterates between two alternating steps and produces a sequence of estimates $\hat{\boldsymbol{\Theta}}^{(t)}$, $t = 0, 1, 2, \ldots$ We start with the E-step given the current parameter estimates $\hat{\boldsymbol{\Theta}}^{(t)}$.

E-step

In this step, we impute values for the unobserved $\Delta_{i,k}$ by

$$\widehat{\Delta}_{i,k}^{(t+1)}=\mathrm{E}(\Delta_{i,k}|\boldsymbol{X},\widehat{\boldsymbol{\Theta}}^{(t)})=\Pr(\Delta_{i,k}=1|\boldsymbol{X},\widehat{\boldsymbol{\Theta}}^{(t)})=\frac{\widehat{w}_k^{(t)}\phi(\boldsymbol{x}_i;\widehat{\mu}_k^{(t)},\widehat{\sum}^{(t)})}{\sum_{k'=1}^{K}\widehat{w}_{k'}^{(t)}\phi(\boldsymbol{x}_i;\widehat{\mu}_{k'}^{(t)},\widehat{\sum}^{(t)})}. \tag{11}$$

Plugging them into (10), we obtain the so-called penalized Q-function:

$$Q_p(\boldsymbol{\Theta},\widehat{\boldsymbol{\Theta}}^{(t)})=\sum_{i=1}^{n}\sum_{k=1}^{K}\widehat{\Delta}_{i,k}^{(t+1)}\{\log w_k+\log \phi(\boldsymbol{x}_i;\mu_k, \sum)\}-\lambda\sum_{j=1}^{p}\sum_{1\le k<k'\le K}\tau_{k,k'}^{(j)}\left|\mu_{k,j}-\mu_{k',j}\right|.$$

M-step

The goal is to update the parameter estimates via

$$\widehat{\boldsymbol{\Theta}}^{(t+1)}=\arg\max_{\boldsymbol{\Theta}} Q_p(\boldsymbol{\Theta},\widehat{\boldsymbol{\Theta}}^{(t)}). \tag{12}$$

Specifically,

$$\frac{\partial Q_p}{\partial w_k}=0 \Rightarrow \widehat{w}_k^{(t+1)}=\frac{1}{n}\sum_{i=1}^{n}\widehat{\Delta}_{i,j}^{(t+1)} \tag{13}$$

$$\frac{\partial Q_p}{\partial \sigma_j^2}=0 \Rightarrow (\widehat{\sigma}_j^{(t+1)})^2=\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\widehat{\Delta}_{i,k}^{(t+1)}(x_{i,j}-\widehat{\mu}_{k,j}^{(t)})^2, 1\le j\le p, \tag{14}$$

and

$$\widehat{\mu}^{(t+1)}=\arg\min_{\mu}\frac{1}{2}\sum_{i=1}^{n}\sum_{k=1}^{K}\left\{\widehat{\Delta}_{i,k}^{(t+1)}\sum_{j=1}^{p}\frac{(x_{i,j}-\mu_{k,j})^2}{(\widehat{\sigma}_j^{(t)})^2}\right\}+\lambda\sum_{j=1}^{p}\sum_{1\le k<k'\le K}\tau_{k,k'}^{(j)}\left|\mu_{k,j}-\mu_{k',j}\right| \tag{15}$$

The optimization of (15) is nontrivial and is discussed in detail next.

Estimation of the cluster means

In general, objective function (15) can be transformed into a quadratic programming problem, and solved by a commercially available package. This approach, however, can be iniefficient in practice, especially for a large number of variables *p*. Thus, we propose a more efficient iterative algorithm based on the standard local quadratic approximation (Fan and Li, 2001). Local quadratic approximation has been used in a number of variable selection procedures and its convergence properties have been studied by Fan and Li (2001) and Hunter and Li (2005). Specifically, we approximate

$$|\mu_{k,j}^{(s+1)}-\mu_{k',j}^{(s+1)}|\approx\frac{(\mu_{k,j}^{(s+1)}-\mu_{k',j}^{(s+1)})^2}{2|\widehat{\mu}_{k,j}^{(s)}-\widehat{\mu}_{k',j}^{(s)}|}+\frac{1}{2}|\widehat{\mu}_{k,j}^{(s)}-\widehat{\mu}_{k',j}^{(s)}|, \tag{16}$$

where *s* is the iteration index (different from *t*, which is used to denote different iterations of the EM algorithm, whereas *s* is used to denote iterations of the local quadratic approximation within the M-step), and $\hat{\mu}^{(s)}$ are the estimates from the previous iteration. This approximation converts the minimization in (15) into a generalized ridge (quadratic) problem, which can be solved in closed form. For example, for each *j* (notice that (15) can be decomposed into *p* separate minimization problems), we solve (iteratively over *s*)

$$\min_{\mu_{(j)}^{(s+1)}}\frac{1}{2(\widehat{\sigma}_j^{(t)})^2}\sum_{i=1}^{n}\sum_{k=1}^{K}\widehat{\Delta}_{i,k}^{(t+1)}(x_{i,j}-\mu_{k,j}^{(s+1)})^2+\lambda\sum_{1\le k<k'\le K}\tau_{k,k'}^{(j)}\frac{(\mu_{k,j}^{(s+1)}-\mu_{k',j}^{(s+1)})^2}{2\left|\widehat{\mu}_{k,j}^{(s)}-\widehat{\mu}_{k',j}^{(s)}\right|}. \tag{17}$$

For numerical stability, we threshold the absolute value of $\widehat{\mu}_{k,j}^{(s)}-\widehat{\mu}_{k',j}^{(s)}$ at a lower bound of 10^{-10}, and at the end of the iterations, set all estimates equal to 10^{-10} to zero.

We note that the M-step of maximizing the penalized *Q*-function does not have closed form solutions, and its maximizer is obtained 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 maximizing the penalized *Q*-function over $\boldsymbol{\Theta}$, but with some of its elements fixed at their previous values. By Theorem 3 in Meng and Rubin (1993), our algorithm is guaranteed to converge to a stationary point.

4. Numerical Results

In this section, we illustrate the performance of the proposed pairwise variable selection method on three synthetic examples with four clusters for Simulations 1 and 3 and five clusters for Simulation 2. We compare four methods: Gaussian mixture model-based clustering without a penalty, the adaptive ℓ_1 penalty (7), the adaptive ℓ_∞ (8) and our proposed adaptive pairwise fusion penalty (6). We refer to them as “GMM”, “AL1”, “ALP”

and "APFP" respectively. The non-adaptive PFP method was also applied and is generally dominated by APFP; its results are omitted for space considerations. In Simulations 1 and 2, the same number of observations, i.e., 20, are generated from each cluster, while in Simulation 3, we generate different number of observations for different clusters. The number of clusters K and the tuning parameter λ are selected using the BIC criterion, as described in Section 2.3. For benchmarking purposes, we also calculate the solution by specifying the true number of clusters, namely $K = 4$ for Simulations 1 and 3 and $K = 5$ for Simulation 2, and only select λ using BIC. We repeat this 50 times for each simulation and record the average clustering error rates as compared to the true cluster labels, and average selection rate for both informative and non-informative variables. To compute the clustering error rates, the predicted class labels are calculated by a majority vote, i.e., if most data points in a particular predicted cluster belong to a true cluster k ($1 \leq k \leq K$), then all data points in this predicted cluster are labeled as k.

The performance of the EM algorithm in model-based clustering depends on the choice of the initial values for the parameters since the likelihood function is not convex, and the algorithm can only converge to a local maximum. To get a good starting value, we first fit 100 GMMs (without penalty) with different random initial values, and use the estimate with the highest likelihood as a starting value for the EM algorithm. In our simulations, the EM algorithm usually converged after about 100 iterations.

Simulation 1

In this scenario, there are four clusters and $\mathbf{p} = 220$, with the first 20 being informative and the remaining ones non-informative. The variables were generated according to the following mechanism: the first 20 are independently distributed $N(\mu_{k,j}, \sigma^2)$ for cluster k, whereas the remaining 200 variables are all i.i.d. $N(0, 1)$ for all four clusters. Table 1 gives the means for the first 20 variables. For example, in cluster 1, variables 1–10 all have the same mean value 2.5, and variables 11–20 all have the same mean value 1.5. Figure 2 (left panel) illustrates the distribution of the informative variables. Notice that variables 1–10 are non-informative for separating clusters 2 and 3, while variables 11–20 are non-informative for separating clusters 1 and 2 (as well as clusters 3 and 4). We consider two values of the common variance, $\sigma^2 = 1$ and $\sigma^2 = 4$. The former creates a high "signal-to-noise ratio (SNR)" scenario, while the latter simulates a situation where the "signal-to-noise ratio" is low.

Simulation 2

A five cluster scenario is considered. There are a total of $p = 230$ variables with the first 30 informative and the other 200 non-informative. Similarly to Simulation 1, the informative variables are independently distributed as $N(\mu_{k,j}, \sigma^2)$ for cluster k, whereas the remaining 200 variables are all i.i.d. $N(0, 1)$ for all five clusters. Table 1 gives the mean values for the informative variables, and Figure 2 (right panel) illustrates the distribution of the informative variables. Notice that variables 1–10 are non-informative for separating clusters 1 and 2, as well as clusters 3 and 4; variables 11–20 are non-informative for separating clusters 2, 3 and 4; and variables 21–30 are non-informative for separating clusters 2 and 3, as well as clusters 4 and 5. We, again, consider $\sigma^2 = 1$ (high signal-to-noise ratio) and $\sigma^2 = 4$ (low signal-to-noise ratio).

Simulation 3

This simulation is designed to test the proposed method on unbalanced data, i.e., data where clusters have different sample sizes. All the settings in this simulation are the same as in Simulation 1 (high SNR), except that the sample size for clusters 3 and 4 has been increased

to 200. Therefore, there are two small clusters (1 and 2) with 20 observations each and two large clusters (3 and 4) with 200 observations each.

The results over 50 replications for all simulation scenarios are summarized in Table 2. When the signal-to-noise ratio in Simulations 1 and 2 is high, all four methods select the correct number of clusters and the error rates are very close to zero. On the other hand, in the low signal-to-noise ratio setting, GMM and ALP completely fail to select the correct number of clusters, and have a high error rate. The performance of the AL1 and APFP methods also degrade, but both are still able to select the correct number of clusters most of the time. Further, the error rate of the APFP method is comparable with that of the AL1 method. In terms of variable selection, AL1, ALP and APFP are able to identify the informative variables, but APFP is more effective than ALP and AL1 at removing non-informative variables. The results for Simulation 3 are very similar to those of Simulation 1 with high SNR, which shows that unbalanced data do not affect performance of any of the methods.

If a variable is non-informative for separating a pair of clusters, and the corresponding estimated means are also the same, we consider this correct "fusion". Table 3 summarizes these results. Specifically, each row in the table gives the proportion of correctly fused variables (average over 50 replications) out of the ten that are non-informative for separating the corresponding pair of clusters (indicated in the third column). For example, the first row shows that for the APFP method, on average 91.6% of the variables among the first ten are correctly fused for clusters 2 and 3. It is also clear that APFP dominates both AL1 and ALP in terms of correctly fusing the cluster means. Although AL1 and ALP can correctly fuse some cluster means (e.g., in the first and second row), these results are artifacts. For example, in Simulation 1, the means of clusters 2 and 3 for variables 1–10 are all equal to zero, which happens to be the value that the ℓ_1 penalty shrinks to. The same reasoning applies to clusters 2, 3 and 4 for variables 11–20 in Simulation 2. On the other hand, in Simulation 1, although clusters 1 and 2 (as well as clusters 3 and 4) have the same mean value for variables 11–20, the AL1 method fails to fuse them, since their mean value is different from zero. The ALP method only shrinks the cluster mean with the largest magnitude, such as the means of clusters 1 and 2 and cluster 3 and 4 for variables 11–20 in Simulation 1. We can also see that both AL1 and ALP are unable to perform pairwise variable selection for unbalanced clusters in Simulation 3. In contrast to Simulation 1, the overall sample mean in Simulation 3 (red star in Figure 3) does not lie at the centroid of the four cluster means. This explains why AL1 fails to identify non-separable clusters 2 and 3 for variables 11–20 and ALP fails to identify non-separable clusters 3 and 4, which they were able to identify in Simulation 1. The APFP method identifies the correct structure in all these scenarios.

5. Applications to Gene Expression Data

In this section, we apply the pairwise fusion method to two gene microarray data sets. To illustrate the method, we pre-select a subset of genes from each data by ranking the genes according to their variance and only using the top 100 and bottom 100 genes. We anticipate that high variance genes are more informative than low variance genes for clustering purposes, although, as the results below show, this is not always true. Notice that selection does not use any class label information. The obtained 200 variables (genes) are centered before clustering.

5.1 The SRBCT Data

This data set contains the expression profiles of 2308 genes, obtained from 83 tissue samples of small round blue cell tumors (SRBCT) of childhood cancer (Khan et al., 2001). The 83

samples are classified into four tumor subtypes: Ewing's sarcoma (EWS), rhabdomyosarcoma (RMS), neuroblastoma (NB), and Burkitt's lymphoma (BL).

The results in Table 4 (SRBCT) show that all these methods select six clusters via BIC and produce the same error rate of 1.4%. Table 5 shows the confusion matrix for the APFP method. Each row corresponds to a tumor subtype, and each column to an identified cluster. It can be seen that subtype EWS is split into clusters 2 and 6, and subtype RMS into clusters 1 and 3. This result suggests possible existence of heterogeneous structures within these two subtypes.

From Table 4, we can also see that both GMM and ALP select all 200 genes, while APFP selects 92 from the top 100 genes and 66 from the bottom 100 genes, and AL1 selects all top 100 genes and 88 from the bottom 100 genes. This is a somewhat unexpected result. To further investigate this issue, two F-statistics and their p-values were computed for each gene; the first one compares the four tumor subtypes, while the second one the six identified clusters. The results are shown in Figure 4. Notice that although genes with a large variance tend to be informative (since they tend to have small p-values as shown in the left panels of Figure 4), genes with a small variance are not necessarily non-informative for clustering. The right panels in Figure 4 show that among the bottom 100 genes by variance there is a number of genes with relatively small p-values, both for discriminating the true subtypes and the found clusters. These turn out to be the genes that are selected by the APFP method from the bottom 100 genes. Further, the left panels in Figure 4 show that some of top 100 genes have large p-values. Indeed, the four genes that have the largest p-values are not selected by APFP. Overall, Figure 4 provides insight into why 66 genes are selected by the APFP method from the bottom 100 group, and why some of the genes in the top 100 group are not selected. The selection of all the genes by the L1 method is obviously not satisfactory.

Figure 5 shows the results for pairwise fusion. The rows correspond to the 92 (out of top 100) genes selected by the APFP method and the column to pairs of clusters. There are a total of 15 pairs formed from the six identified clusters. A black (white) spot indicates that the estimated means of the corresponding gene for the two clusters are different (the same). For example, the gene with ID "435953" is non-informative for separating clusters 1 and 3, as well as clusters 2 and 5, and clusters 4 and 6. It can be seen that most genes are informative for only a subset of clusters. Compared to the "one-in-all-out" approach, this result is more informative for describing the functions of a gene with respect to discriminating different tumor subtypes.

5.2 PALL Data Set

This data set contains gene expression profiles for 12,625 genes from 248 patients (samples) with pediatric acute lymphoblastic leukemia (PALL), see Yeoh et al. (2002) for more details. The samples are classified into six tumor subtypes: T-ALL (43 cases), E2A-PBX1 (27 cases), TEL-AML (79 cases), hyperdiploid>50 (64 cases), BCR-ABL (15 cases) and MLL (20 cases). The original data had a large number of missing intensities and the following pre-processing was applied. All intensity values less than one were set to one; then all intensities were transformed to log-scale. Further, all genes with log-intensities equal to zero for more than 80% of the samples were discarded, thus leaving 12,083 genes for further consideration. From the pre-processed data, the top and bottom 100 genes were selected according to the overall variance criterion described above. All variables were centered.

From Table 4 (PALL), we can see that GMM, AL1 and APFP methods select 12, 7 and 9 clusters, respectively, and produce comparable error rates (25%~27%), all of which are

significantly lower than that of ALP (41.1%). Table 6 shows the confusion matrix for the APFP method. Unlike the results on the SRBCT data, the clusters discovered by APFP are generally not consistent with the six subtypes. However, subtypes E2A-PBX1 and T-ALL are largely captured by clusters 3 and 7, most samples in subtype hyperdiploid>50 are assigned to clusters 4 and 6, while TEL-AML is split amongst clusters 1, 2 and 9. This result suggests the possible presence of a more complex structure in some of the subtypes.

Figure 6 shows the scatter plot of variance vs *p*-values obtained from the two *F*-statistics as described above. Once again, genes with a large variance do not necessarily correspond to small *p*-values, and vice versa. Figure 7 provides a detailed illustration of the gene functions with respect to discriminating different tumor subtypes.

6. Conclusions

We have developed a method for simultaneously clustering high-dimensional data and selecting informative variables, by employing a penalized model-based clustering framework. In particular, the proposed method penalizes the difference between the cluster means for each pair of clusters and for each variable, which allows one to identify and remove non-informative variables for selected subsets of clusters. This allows to gain more insight into the function of particular variables and potentially discover heterogeneous structures that other available methods are unable to capture. Our numerical work suggests that this penalty proves more effective in removing non-informative variables than an ℓ_1 penalty method, and provides better interpretation. Possible extensions include allowing for different variances and fusing variances as well as the means, as discussed at the start of Section 2.1, as well as extensions to non-Gaussian data. Applications to problems other than clustering are another possibility; a similar penalty for simultaneously selecting factors and collapsing levels in ANOVA was proposed by Bondell and Reich (2009) while this paper was under review.

Acknowledgments

We thank an Associate Editor and two referees for helpful suggestions. E. Levina's research is partially supported by NSF grants DMS-0505424 and DMS-0805798 and a Rackham faculty grant. G. Michailidis's research is partially supported by NIH grant 5P 41RR018627 and MEDC grant GR-687. J. Zhu's research is partially supported by NSF grants DMS-0705532 and DMS-0748389.

References

Banfield J, Raftery A. Model-based Gaussian and non-Gaussian clustering. Biometrics. 1993; 49:803–821.

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

Bondell H, Reich B. Simultaneous factor selction and collapsing levels in ANOVA. Biometrics. 2009; 65:169–177. [PubMed: 18510652]

Fan J, Li R. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American Statistical Asscociation. 2001; 96:1348–1360.

Fraley C. Algorithms for model-based Gaussian hierarchical clustering. SIAM Journal on Scientific Computing. 1993; 20:270–281.

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

Gordon A. A review of hierarchical classification. Journal of the Royal Statistical Society, Series A. 2008; 150:119–137.

Hoff P. Model-based subspace clustering. Bayesian Analysis. 2006; 1:321–344.

Hunter D, Li R. Variable selection using MM algorithms. Annals of Statistics. 2005; 33:1617–1642. [PubMed: 19458786]

Jornsten R, Keles S. Mixture models with multiple levels, with application to the analysis of multifactor gene expression data. Biostatistics. 2008; 9:540–554. [PubMed: 18256042]

Kaufman, L.; Rousseeuw, P. Finding groups in data: an introduction to cluster analysis. New York: John Wiley & Sons; 1990.

Khan J, Wei JS, Ringner M, Saal LH, Ladanyi M, Westermann F, Berthold F, Schwab M, Antonescu CR, Peterson C, Meltzer PS. Classification and diagnostic prediction of cancers using gene expression profiling andartificial neural networks. Nature Medicine. 2001; 7:673–679.

MacQueen, J. Some methods for classification and analysis of multivariate observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability; University of California Press; Berkeley. 1967. p. 281-297.

McLachlan, G.; Basford, K. Mixture models: inference and applications to clustering. New York: Marcel Dekker; 1988.

McLachlan, G.; Peel, D. Finite mixture models. New York: John Wiley & Sons; 2002.

Meng XL, Rubin D. Maximum likelihood estimation via the ECM algorithm: a general framework. Biometrika. 1993; 80:267–278.

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

Parsons, L.; Haque, E.; Liu, H. Evaluating subspace clustering algorithms; SIAM International Conference on Data Mining; SIAM. 2004. p. 48-56.

Raftery A, Dean N. Variable selection for model-based clustering. Journal of the American Statistical Asscociation. 2006; 101:168–178.

Tadesse MG, Sha N, Vannucci M. Bayesian variable selection in clustering high-dimensional data. Journal of the American Statistical Asscociation. 2005; 100:602–617.

Tibshirani R, Saunders M, Rosset S, Zhu J, Knight K. Sparsity and smoothness via the fused lasso. Journal of the Royal Statistical Society, Series B. 2005; 67:91–108.

Wang S, Zhu J. Variable selection for model-based high-dimensional clustering and its application to microarray data. Biometrics. 2007; 64:440–448. [PubMed: 17970821]

Xie B, Pan W, Shen X. Penalized model-based clustering with cluster-specific diagonal covariance matrices and grouped variables. Electronic Journal of Statistics. 2008; 2:168–212. [PubMed: 19920875]

Yeoh E-J, Ross M, Shurtleff S, Williams W, Patel D, Mahfouz R, Behm F, Raimondi S, Relling M, Patel A, Cheng C, Campana D, Wilkins D, Zhou X, Li J, Liu H, Pui C-H, Evans W, Naeve C, Wong L, Downing J. Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling. Cancer Cell. 2002; 1:133–143. [PubMed: 12086872]

Zou H. The adaptive LASSO and its oracle properties. Journal of the American Statistical Asscociation. 2006; 101:1418–1429.

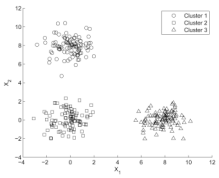

Figure 1.
A toy example. Variable 1 is informative for separating clusters 2 and 3, and variable 2 is informative for separating clusters 1 and 2.

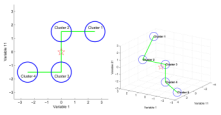

Figure 2.
The distribution of informative variables in Simulation 1 (left) and Simulation 2 (right). The red star indicates the position of the overall sample mean.

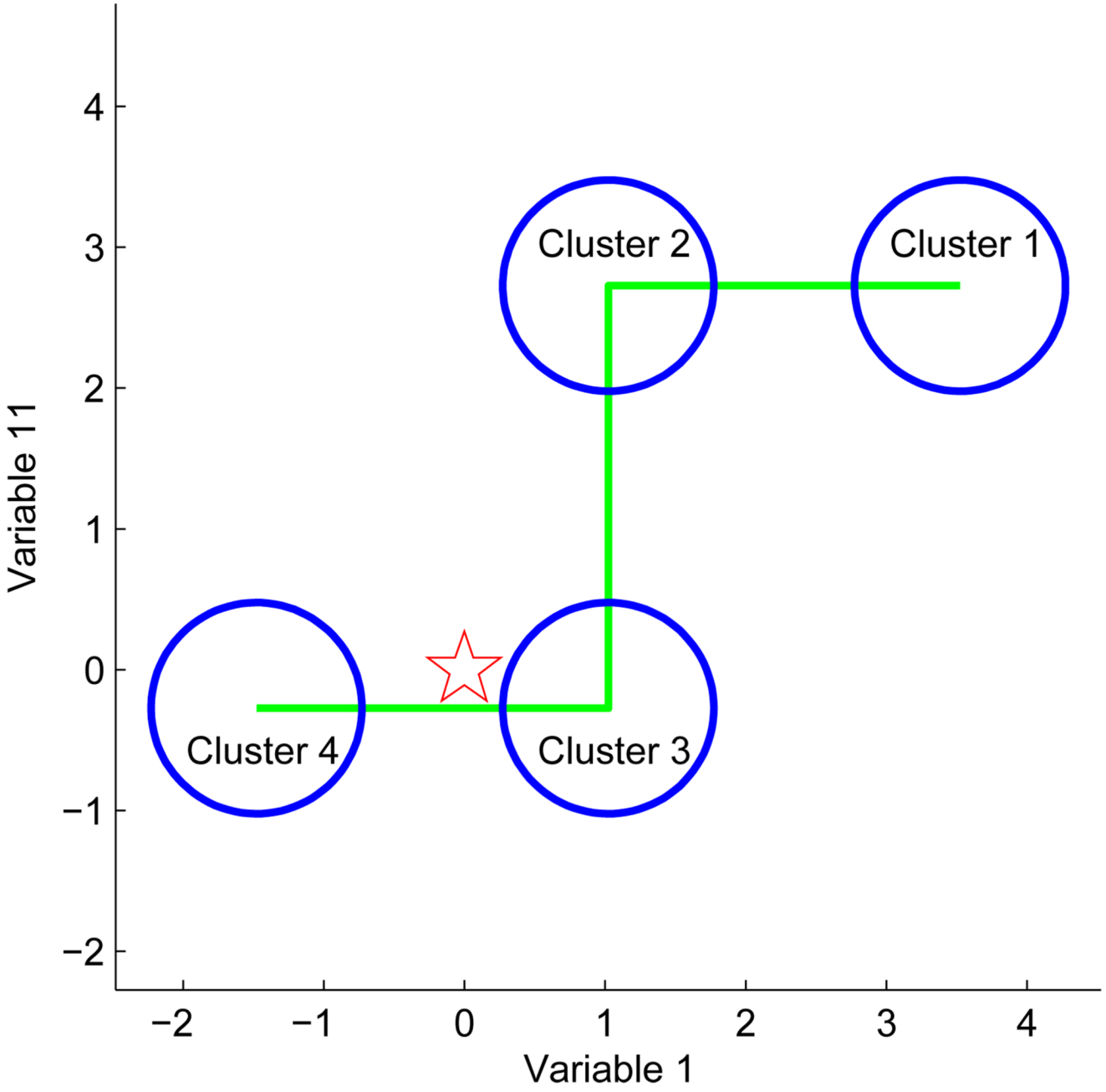

Figure 3.
Simulation 3. The sample sizes of clusters 1, 2, 3 and 4 are 20, 20, 200, and 200, respectively. The red star indicates the position of the overall sample mean, and the plot is shifted to show centered data.

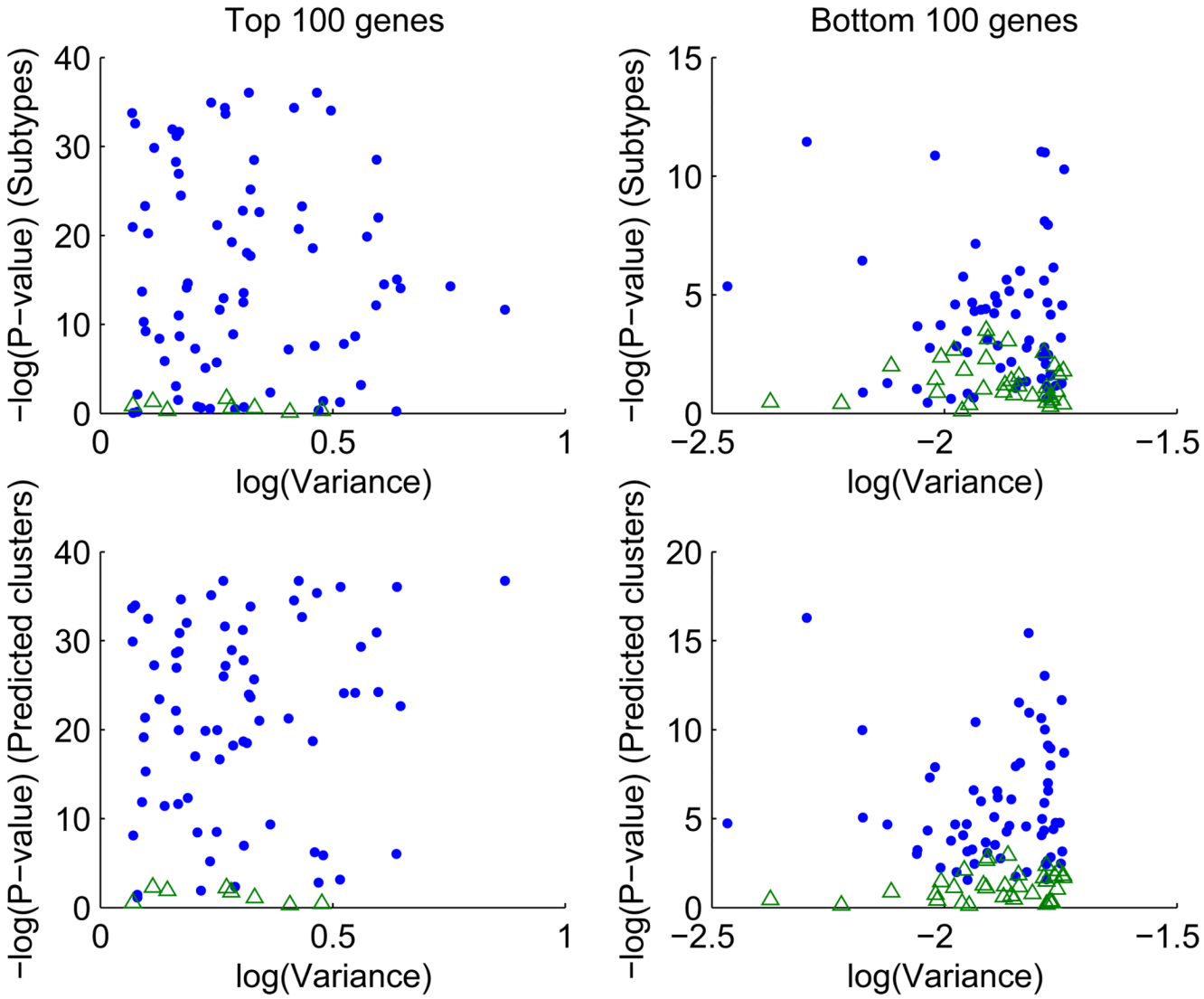

Figure 4.
Plots of the negative logarithm *p*-values vs variance for SRBCT data. The left column is the top 100 genes (largest overall variances), and the right column is the bottom 100 genes. The upper row is negative logarithm *p*-values corresponding to an *F*-statistics comparing four tumor subtypes, and the lower row is the negative logarithm *p*-values for the six identified clusters. Triangles denote the genes that are not selected by the APFP method.

Figure 5.
Pairwise variable selection results for the APFP method on the SRBCT data with top 100 genes. Each row corresponds to a gene. Each column corresponds to a cluster pair; for example, “1/2” indicates clusters 1 and 2. A black (white) spot indicates that the estimated means of the corresponding gene for the two clusters are different (the same). For example, gene “435953” is non-informative for separating clusters 1 and 3, 2 and 5, and 4 and 6.

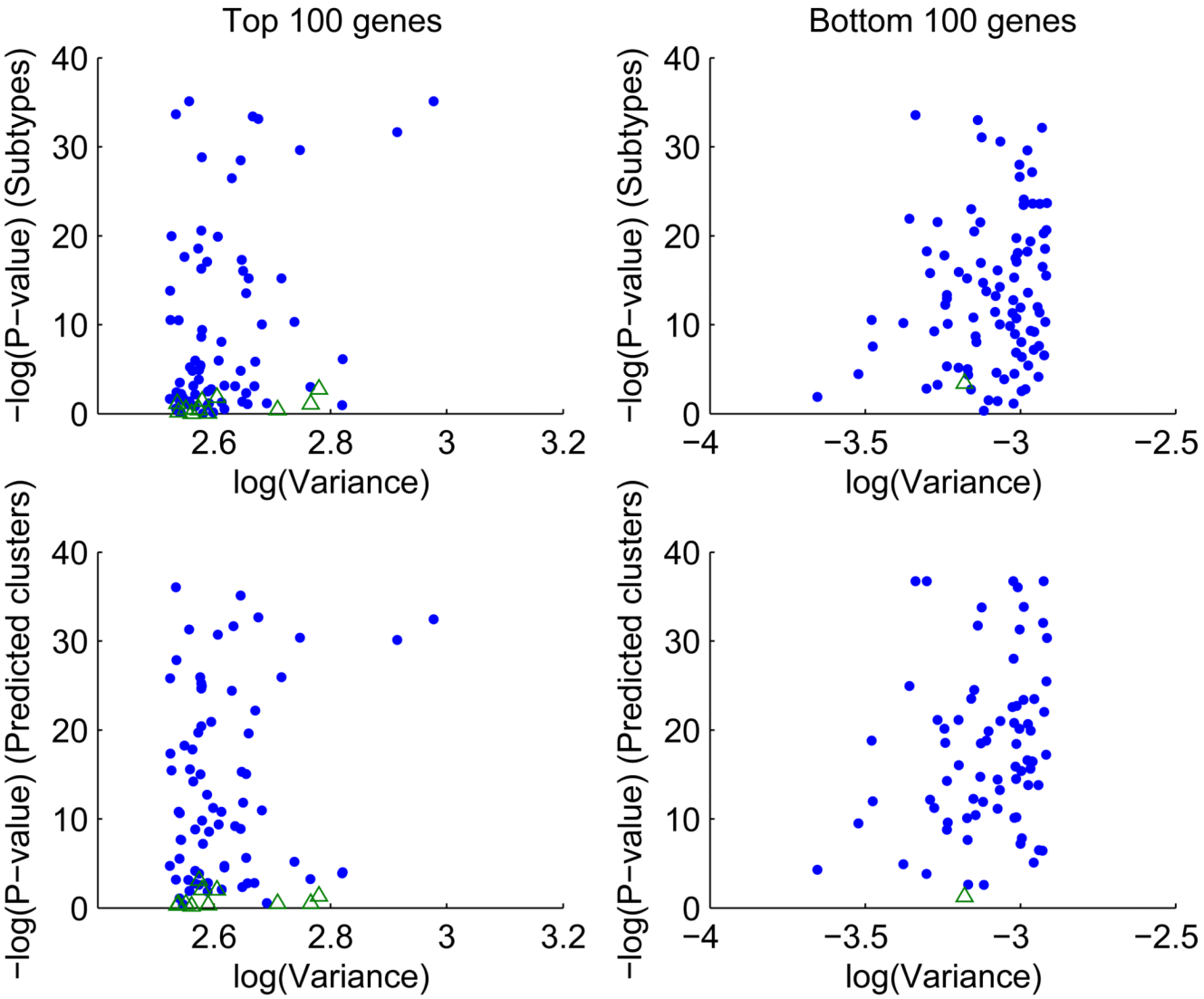

Figure 6.
Plots of the negative logarithm *p*-values vs variance for PALL data. The left column is the top 100 genes (largest overall variances), and the right column is the bottom 100 genes. The upper row is negative logarithm *p*-values corresponding to an *F*-statistics comparing four tumor subtypes, and the lower row is the negative logarithm *p*-values for the six identified clusters. Triangles denote the genes that are not selected by the APFP method.

Figure 7.
Pairwise variable selection results for the APFP method on the PALL data with top 100 genes. Each row corresponds to a gene. Each column corresponds to a cluster pair; for example, “1/2” indicates clusters 1 and 2. A black (white) spot indicates that the estimated means of the corresponding gene for the two clusters are different (the same).

Table 1

Means of informative variables in Simulations 1–3.

Simulation	Variable	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Cluster 5
1 & 3	1–10	2.5	0	0	−2.5	—
	11–20	1.5	1.5	−1.5	−1.5	—
2	1–10	2.5	2.5	0	0	−2.5
	11–20	−2.5	0	0	0	2.5
	21–30	2.5	0	0	−2.5	−2.5

Table 2

Prediction and variable selection results for Simulations 1–3. Each table cell gives average(SD) over 50 repetitions. “K” is the average number of selected clusters, “ER” is the average clustering error rate, “ER (correct K)” is the average error rate when K is set to the true value rather than selected by BIC, “Info” is the average proportion of selected informative variables, and “Noninfo” is the average proportion of selected non-informative variables. “High SNR” corresponds to $\sigma^2 = 1$, and “Low SNR” corresponds to $\sigma^2 = 4$.

Sim. (SNR)	Method	*K*	ER (%)	ER (correct K)	Info (%)	Noninfo (%)
1 (High)	GMM	3 (0)	25 (0)	0 (0)	100 (100)	100 (100)
	AL1	4 (0)	0 (0)	0 (0)	100 (100)	7.1 (7.1)
	ALP	4 (0)	0 (0)	0 (0)	100 (100)	2.4 (2.4)
	APFP	4 (0)	0 (0)	0 (0)	100 (100)	0.5 (0.5)
1 (Low)	GMM	3 (0)	33 (4.9)	20.6 (8.5)	100 (100)	100 (100)
	AL1	3.8 (0.6)	19.2 (14.9)	14.2 (10.7)	100 (100)	6 (6)
	ALP	3 (0)	34.1 (14.5)	14.4 (14)	95.9 (95.9)	4 (4)
	APFP	3.7 (0.6)	19.2 (16.7)	15.1 (12.6)	100 (100)	2.3 (2.3)
2 (High)	GMM	3 (0)	40 (0)	0 (0.2)	100 (100)	100 (100)
	AL1	5 (0)	0 (0)	0 (0)	100 (100)	6.9 (6.9)
	ALP	5 (0)	0 (0.1)	0 (0.1)	100 (100)	1.8 (1.8)
	APFP	5 (0)	0 (0)	0 (0)	100 (100)	1.1 (1.1)
2 (Low)	GMM	3 (0)	40.3 (0.7)	15.3 (5.3)	100 (100)	100 (100)
	AL1	4.7 (0.6)	11.7 (9.8)	8.3 (5.3)	100 (100)	10 (10)
	ALP	3 (0)	40.1 (0.4)	5.8 (3)	100 (100)	5.2 (5.2)
	APFP	4.7 (0.5)	11.7 (7.7)	9.2 (5.5)	100 (100)	2.4 (2.4)
3	GMM	3 (0)	4.5 (0)	0 (0)	100 (100)	100 (100)
	AL1	4 (0)	0 (0)	0 (0)	100 (100)	8.1 (8.1)
	ALP	3.9 (0.2)	0.3 (1.1)	0 (0)	100 (100)	5.9 (5.9)
	APFP	4 (0.1)	0 (0)	0 (0)	100 (100)	0.2 (0.2)

Table 3

Pairwise variable selection results for Simulations 1–3. “Pair” corresponds to non-separable cluster pairs for the variables in the corresponding row. For example, the first row indicates that variables 1–10 are non-informative for separating clusters 2 and 3. The numbers in the following columns show what proportion of variables of the set are identified as non-informative for separating a given pair of clusters by each method. The optimal value is 10 in each case. All results are averages (SDs) over 50 repetitions.

Sim. (SNR)	Variables	Pair	AL1(%)	ALP(%)	APFP(%)
1 (High)	1–10	2/3	96.6 (5.2)	0.2 (1.4)	91.6 (9.1)
	11–20	1/2	0.2 (1.4)	40.8 (18.9)	91.8 (8.5)
		3/4	0 (0)	42.2 (21.4)	92.2 (7.9)
1 (Low)	1–10	2/3	95.6 (9.3)	6 (21.4)	79.8 (17.6)
	11–20	1/2	1 (3.0)	85 (16.2)	78.2 (21.2)
		3/4	0.4 (2.0)	79.6 (14.1)	84 (13.4)
2 (High)	1–10	1/2	0.2 (1.41)	0.2 (1.41)	84.2 (12.3)
		3/4	34.6 (28.1)	0.4 (2.0)	87.4 (9.7)
	11–20	2/3	98 (5.0)	0.2 (1.4)	94 (8.1)
		2/4	97.6 (4.8)	0.4 (2.0)	93.4 (8.2)
		3/4	97.2 (4.5)	0.2 (1.4)	93.2 (8.9)
	21–30	2/3	30.2 (30.1)	0.4 (2.0)	83.8 (12.1)
		4/5	0 (0)	0 (0)	88.2 (10.6)
2 (Low)	1–10	1/2	0.2 (1.41)	17 (10.9)	72.4 (17)
		3/4	73 (14.7)	0 (0)	74.4 (18.5)
	11–20	2/3	94.8 (6.46)	0 (0)	89.2 (11.2)
		2/4	95.4 (5.4)	0 (0)	89.4 (9.8)
		3/4	95.4 (6.1)	0 (0)	89 (10.2)
	21–30	2/3	76.8 (14.9)	0 (0)	67.8 (21.8)
		4/5	0 (0)	21.2 (13.8)	74.4 (16.8)
	1–10	2/3	0.2 (1.4)	0.4 (2.0)	94.6 (6.8)

Sim. (SNR)	Variables	Pair	AL1(%)	ALP(%)	APFP(%)
3	11–20	1/2	0.2 (1.4)	60.8 (14.7)	92.6 (6.6)
		3/4	0 (0)	0 (0)	96.8 (6.2)

Table 4

Clustering results for the SRBCT and PALL data sets. “Top 100” and “Bottom 100” correspond to the number of genes that are selected from the top 100 and bottom 100 genes respectively, as ranked by overall variance.

Data	Method	*K*	Error rate (%)	Top 100 (%)	Bottom 100 (%)
SRBCT	GMM	6	1.4	100	100
	AL1	6	1.4	100	88
	ALP	6	1.4	100	100
	APFP	6	1.4	92	66
PALL	GMM	12	25.7	100	100
	AL1	7	24.7	94	100
	ALP	5	41.1	100	100
	APFP	9	27.0	89	99

Table 5

Confusion matrix of the APFP method for the SRBCT data. Rows correspond to tumor subtypes, and columns to identified clusters.

Subtype	C1	C2	C3	C4	C5	C6
EWS	0	18	0	0	0	11
RMS	6	0	9	0	0	0
NB	1	0	0	0	17	0
BL	0	0	0	11	0	0

Table 6

Confusion matrix of the APFP method for the PALL data. Rows correspond to tumor subtypes, and columns to identified clusters.

Subtype	C1	C2	C3	C4	C5	C6	C7	C8	C9
BCR-ABL	0	0	0	2	6	7	0	0	0
E2A-PBX1	0	0	25	0	0	1	0	1	0
hyperdiploid>50	1	1	0	35	0	24	0	2	1
MLL	1	0	2	0	13	0	0	4	0
TEL-AML	30	18	0	0	0	0	0	0	31
T-ALL	0	0	0	0	5	0	33	5	0