A Penalized Matrix Normal Mixture Model for Clustering Matrix Data

Jinwon Heo and Jangsun Baek *

Department of Mathematics and Statistics, Chonnam National University, 77 Yongbong-ro, Buk-gu, Gwangju 61186, Korea; jwwww0124@naver.com
* Correspondence: jbaek@jnu.ac.kr; Tel.: +82-62-530-3446

Abstract: Along with advances in technology, matrix data, such as medical/industrial images, have emerged in many practical fields. These data usually have high dimensions and are not easy to cluster due to their intrinsic correlated structure among rows and columns. Most approaches convert matrix data to multi-dimensional vectors and apply conventional clustering methods to them, and thus, suffer from an extreme high-dimensionality problem as well as a lack of interpretability of the correlated structure among row/column variables. Recently, a regularized model was proposed for clustering matrix-valued data by imposing a sparsity structure for the mean signal of each cluster. We extend their approach by regularizing further on the covariance to cope better with the curse of dimensionality for large size images. A penalized matrix normal mixture model with lasso-type penalty terms in both mean and covariance matrices is proposed, and then an expectation maximization algorithm is developed to estimate the parameters. The proposed method has the competence of both parsimonious modeling and reflecting the proper conditional correlation structure. The estimators are consistent, and their limiting distributions are derived. We applied the proposed method to simulated data as well as real datasets and measured its clustering performance with the clustering accuracy (ACC) and the adjusted rand index (ARI). The experiment results show that the proposed method performed better with higher ACC and ARI than those of conventional methods.

Keywords: clustering; image analysis; matrix normal distribution; expectation maximization algorithm; penalized likelihood

1. Introduction

Among the clustering methods, the Gaussian mixture model, which is model-based clustering, is widely used thanks to its easy interpretability. Since most clustering approaches suffer from the curse of dimensionality in “high dimension/low sample size data”, so does model-based clustering. The problem is further exacerbated in a Gaussian mixture model that has to estimate multiple covariance matrices. For example, given a $p$-dimensional random sample $\{X_i = (X_{i1}, \ldots, X_{ip})^T, i = 1, \ldots, n\}$ for a multivariate Gaussian mixture model with $G$ components, we need to estimate a total of $Gp(p+1)/2 + Gp + G - 1$ parameters. So, the number of parameters to be estimated grows quickly with dimension $p$. To address this problem, a penalized Gaussian mixture clustering model has been proposed that imposes constraints on the mean or precision matrix. In particular, Ref. [1] suggests a penalized multivariate normal mixture clustering model for variable selection by regularizing the mean structure. Additionally, Ref. [2] suggests a penalized multivariate normal mixture clustering model for variable selection by regularizing the mean structure. In each cluster, there may be some variables that are not relevant to distinguishing it from other clusters. So, finding the non-informative (noise) variables and removing them may largely improve interpretability.

1. Introduction

Among the clustering methods, the Gaussian mixture model, which is model-based clustering, is widely used thanks to its easy interpretability. Since most clustering approaches suffer from the curse of dimensionality in “high dimension/low sample size data”, so does model-based clustering. The problem is further exacerbated in a Gaussian mixture model that has to estimate multiple covariance matrices. For example, given a $p$-dimensional random sample $\{X_i = (X_{i1}, \ldots, X_{ip})^T, i = 1, \ldots, n\}$ for a multivariate Gaussian mixture model with $G$ components, we need to estimate a total of $Gp(p+1)/2 + Gp + G - 1$ parameters. So, the number of parameters to be estimated grows quickly with dimension $p$. To address this problem, a penalized Gaussian mixture clustering model has been proposed that imposes constraints on the mean or precision matrix. In particular, Ref. [1] suggests a penalized multivariate normal mixture clustering model for variable selection by regularizing the mean structure. In each cluster, there may be some variables that are not relevant to distinguishing it from other clusters. So, finding the non-informative (noise) variables and removing them may largely improve interpretability. Additionally, Ref. [2] suggests a penalized multivariate normal mixture clustering model for achieving better clustering performance by regularizing the precision matrices. In another way, Ref. [3] adopts the idea of a penalized covariance estimation from [4] and applies a lasso-type penalty to the off-diagonal component of the precision matrices.
However, as the dimension $p$ increases, the computational speed extremely slows down, making it difficult to estimate the parameters practically for these methods. In recent years, along with advances in technology, matrix data have emerged in many practical fields. The conventional Gaussian mixture method converts these matrix data into vector data. So, the dimension of the data can be extremely large, i.e., a $p \times q$ matrix will be converted to a vector of $pq$ dimension. In addition, clustering a matrix dataset is difficult because correlations between variables or occasions can change across occasions. To solve the aforementioned issues, Ref. [5] suggests finite mixtures of matrix normal distributions for matrix data. Direct analysis of three-way data structures by using a matrix normal distribution allows considering and estimating correlations between variables and occasions. Recently, the matrix mixture model was extended for various non-normal matrix data: Refs. [6,7] proposed finite mixture of matrix skewed distributions, and [8] introduced two matrix-variate distributions—both elliptical heavy-tailed generalizations of the matrix-variate normal distribution that are used in a finite mixture model. Ref. [9] suggested a penalized matrix normal mixture model by imposing a penalty for the mean matrix. This method can capture the row-wise and column-wise correlation simultaneously and has the ability to find the sparsity nature that is inherent in the signals and images. However, the approach of [9] shows weakness for high dimension/low sample size data because it cannot reduce the number of precision parameters to estimate.

In a multivariate normal distribution, the conditional correlation between two variables, given the other variables, is explicitly determined by the precision matrix. That is, if the $(i,j)$th element of the precision matrix is zero, then the conditional correlation between the $i$th and the $j$th variables given the other variables is zero, and they are conditionally independent. Ref. [10] introduced the idea of covariance selection, which consists of inferring a sparse estimation of the precision matrix and interpreting its sparsity structure as a conditional dependency between variables. A Gaussian graphical model is a network whose values on the vertices follow a centered multivariate normal distribution, and has been applied to analyze high-dimensional data with small sample size through the regularized estimation of the precision matrix [11]. A notable approach for the precision matrix estimation is the graphical lasso algorithm [12].

In this paper, the work of [9] is extended by further regularizing the precision matrix as in the Gaussian graphical model, to cope better with the curse of dimensionality for large size images. Specifically, we propose a penalized matrix normal mixture model with lasso-type penalty terms in both mean and covariance matrices, and then an expectation maximization algorithm is developed to estimate the parameters. The estimators are consistent and their limiting distributions are derived. The experiment results show that the proposed approach attains much improved clustering performance in comparison with conventional methods.

The capability of the proposed model is compared with two popular distance-based methods and two conventional model-based methods in Appendix A Table A1. The major contributions of this research to overcome the limitations of the conventional methods are as follows:

1. Since the data type for the Gaussian mixture model is a vector and K-means and spectral clustering methods use dissimilarity and similarity between the observations as input, these methods do not have the modeling ability for the row-wise/column-wise covariance structure of matrix data (images). On the other hand, both [9] and the proposed model can capture the row-wise and column-wise correlation of image pixels simultaneously by estimating the row covariance $U$ and column covariance $V$, where the covariance matrix $\Sigma$ is the Kronecker product of $U$ and $V$. This capability of modeling separable row-wise/column-wise covariance structure helps for better clustering of image data.

2. Because the proposed model is a model-based clustering method, it has the sound mathematical basis and the interpretability of results with the estimated parameters, whereas centroid-based and similarity-based methods do not.
(3) If variables are measured on different scales and one variable has much larger values than the others, then the first variable will dominate the clustering. Therefore, it is preferred to use a clustering method that is invariant against affine transformations. K-means and similarity-based methods using the Euclidean distance are not invariant against affine transformations, whereas Gaussian mixture models are.

(4) Model-based clustering methods have free parameters to estimate and often have difficulty in the estimation of parameters when the data are of a high dimension but the sample is small. Given \(p \times q\), random matrix data \(X\), and the number of clusters \(G\), the conventional Gaussian mixture model needs to estimate a total of \(Gpq(pq + 1)/2 + Gpq + G - 1\) parameters because it converts the data into a vector of \(pq\) dimension. The method \([9]\) is regularized on the mean matrix, and thus it needs to estimate \(Gp(p + 1)/2 + Gq(q + 1)/2 + Gpq + G - 1 - r\) parameters, where \(r\) is the number of the mean parameters values that are shrunken to zero. Since the proposed method is regularized further on the precision matrix, that is, on the row and column covariances, \(U\) and \(V\), with shrunken elements, it needs to estimate \(G(p + 1)/2 + G(q + 1)/2 + Gpq + G - 1 - r^*\) parameters, where \(r^*\) is the number of the mean and precision parameters values that are shrunken to zero. The range of \(r\) and \(r^*\) is \(0 \leq r \leq Gpq\) and \(r \leq r^* \leq G(p + 1)/2 + G(q + 1)/2 + Gpq\) respectively. Therefore, the proposed approach is a more parsimonious model that has the fewest parameters and is able to cope better with the curse of dimensionality for large size images.

(5) There often exist correlations between the rows and the columns of matrix data. A separable covariance structure of the proposed model can effectively reduce parameters, speed up calculations, and provide practical interpretability. Therefore, the proposed method has the competence of both parsimonious modeling to overcome the curse of dimensionality and reflecting the proper conditional correlation structure for high dimension/low sample size data in comparison with the conventional methods.

The paper is organized as follows. In Section 2, the matrix normal mixture model is explained and then the penalized matrix normal mixture model with the mean and precision matrices is introduced. Next, a new EM algorithm for the precision matrices of the proposed model, a model selection method, k-fold cross-validation, and the asymptotic theory are developed. In Section 3, the proposed method is applied to both simulation data and real data. In the simulation data experiment, the clustering performance is compared with that of the K-means clustering method \([13]\), the spectral clustering method \([14]\), and \([9]\)’s method. The consistency of the estimates for the proposed model is shown, using the averaged spectral norm of the difference (SL) and the averaged Frobenius norm of the difference (FL) \([3]\). In the real data experiment, the clustering performance is also compared with that of the K-means clustering method, the spectral clustering method and the method of \([9]\). In Section 4, we summarize the results and discuss future work. The Appendices A–C provides a comparative overview table and the proofs of the penalized maximum likelihood estimator of the mean matrix for the proposed model and the asymptotic theory.

2. Method

A mixture model using matrix normal distribution was previously introduced by \([5]\). The matrix normal mixture model \([15–17]\) is reviewed is shown in Section 2.1. Then, our penalized matrix normal mixture model, estimation, asymptotic theory, and model selection are presented in Sections 2.2–2.4.

2.1. Matrix Normal Mixture Model (MNMM)

2.1.1. Definition of MNMM

The \(p \times q\) random matrix \(X\) has a matrix normal distribution, with mean matrix \(M\) and row covariance \(U\) and column covariance \(V\), denoted by \(X \sim N_{p,q}(M, U, V)\), if the p.d.f. of \(X\) is given by the following:
\[ f(X|M, U, V) = (2\pi)^{-\frac{pq}{2}} |V|^{-\frac{p}{2}} |U|^{-\frac{q}{2}} \exp \left\{ -\frac{1}{2} \text{tr}(V^{-1}(X - M)^T U^{-1}(X - M)) \right\}, \tag{1} \]

where \( M \in \mathbb{R}^{p \times q}, U \in \mathbb{R}^{p \times p}, V \in \mathbb{R}^{q \times q} \) and \(|A|, \text{tr}(A)\) are the determinant value and the trace of a matrix \( A \), respectively. The p.d.f. (1) can be converted to the p.d.f. of the vectorization of \( X \) as follows:

\[ \text{vec}(X) \sim N_{pq}(\text{vec}(M), V \otimes U), \]

where \( \text{vec} \) is the vectorization operator and \( \otimes \) is the Kronecker product. The matrix normal distribution is a generalization of the matrix variate, i.e., it can effectively handle three-way data. In particular, the separable covariance structure has an important advantage, which is reducing the number of parameters. For example, given \( p \times q \) random matrix \( X \), the dimension of inseparable covariance matrix \( \Sigma \) is \( pq \times pq \). However, the dimension of separable covariance matrix \( \Sigma = V \otimes U \) is \( p \times p + q \times q \).

2.1.2. Maximum Likelihood Estimation

The log-likelihood function for given i.i.d observations \( X_1, X_2, \ldots, X_n \) of \( N_{pq}(M, U, V) \) is as follows:

\[
L(M, U, V; X) = -\frac{npq}{2} \log 2\pi - \frac{np}{2} \log |V| - \frac{nq}{2} \log |U| - \frac{1}{2} \sum_{i=1}^{n} \text{tr}\left\{ (V^{-1}(X_i - M)^T U^{-1}(X_i - M)) \right\}.
\]

The maximum likelihood estimators (m.l.e.) for \( M, U, V \) are as follows:

\[
\hat{M} = \bar{X},
\]

\[
\hat{U}^{(t+1)} = \frac{1}{np} \sum_{i=1}^{n} (X_i - \bar{X})(\hat{\Sigma}^{(t)})^{-1}(X_i - \bar{X})^T,
\]

\[
\hat{V}^{(t+1)} = \frac{1}{nq} \sum_{i=1}^{n} (X_i - \bar{X})^T(\hat{U}^{(t+1)})^{-1}(X_i - \bar{X}).
\tag{2}
\]

Note that m.l.e., \( \hat{U} \) and \( \hat{V} \) are of the no closed-form solution, and \( a\hat{V} \) and \( b\hat{U} \), which is a constant multiple of \( \hat{U} \) and \( \hat{V} \), are also solutions. However, \( a\hat{V} \otimes b\hat{U} \) must be the same as \( \hat{V} \otimes \hat{U} \) [15]. \( \hat{U} \) and \( \hat{V} \) can be obtained by the iterative algorithm of (2) until the convergence criterion is satisfied. The criterion is set as \( \|\hat{U}^{(t)} - \hat{U}^{(t+1)}\|_F / \|\hat{U}^{(t)}\|_F < 0.01 \) and \( \|\hat{V}^{(t)} - \hat{V}^{(t+1)}\|_F / \|\hat{V}^{(t)}\|_F < 0.01 \).

2.1.3. EM Algorithm of MNMM

Given i.i.d \( p \times q \) observations \( X_1, X_2, \ldots, X_n \) from the mixture of matrix normal distributions with \( G \) groups, the p.d.f. of the matrix normal mixture is as follows:

\[
g(X) = \sum_{j=1}^{G} \pi_j f(X|\Theta_j),
\]

where \( \Theta_j \) is the set of parameters corresponding to the \( j \)th group, i.e., \( \Theta_j = (M_j, U_j, V_j) \), \( \pi_1, \ldots, \pi_G \) are the weights of belonging to the group \( g \), and \( f \) is the matrix normal distribution with mean \( M_j \), row covariance \( U_j \), column covariance \( V_j \). Then, the log-likelihood function of MNMM is as follows:

\[
L(\Theta; X) = \sum_{i=1}^{n} \log \left\{ \sum_{j=1}^{G} \pi_j f(X_i|\Theta_j) \right\}.
\tag{3}
\]
It is difficult to obtain the maximum log-likelihood value by solving the first derivative Equation (3) because the p.d.f is defined as the sum in logarithm. So, the EM algorithm is widely used to estimate parameters by maximizing the conditional expectation for the latent variable \( z \), given the data instead of the log-likelihood function \([18]\).

In the E-step, the posterior probability of the latent variables \( z_{ij} \) is defined by Bayes’ theorem as follows:

\[
\hat{z}_{ij}^{(t)} = \frac{\pi_{ij}^{(t)} f(X_i | \Theta_i^{(t)})}{\sum_{l=1}^{G} \pi_{ij}^{(t)} f(X_i | \Theta_i^{(l)})}.
\] (4)

In the M-step, the parameters, \( \Theta^{(t+1)} \), that maximize the expected conditional log-likelihood given data are found as follows:

\[
\hat{\Theta}^{(t+1)} = \arg\max_{\Theta} \sum_{i=1}^{n} \sum_{j=1}^{G} \hat{z}_{ij}^{(t)} \log \{\pi_{ij} f(X_i | \Theta_i)\}.
\]

We obtain the following estimates by using the partial derivative and some algebra.

\[
\hat{\pi}_{j}^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ij}^{(t)}}{n}, \quad \hat{M}_{j}^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ij}^{(t)} X_i}{\sum_{i=1}^{n} \hat{z}_{ij}^{(t)}},
\]

\[
\hat{U}_{j}^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ij}^{(t)} (X_i - \hat{M}_{j}^{(t+1)}) (\hat{V}_{j}^{(t)})^{-1} (X_i - \hat{M}_{j}^{(t+1)})^T}{q \sum_{i=1}^{n} \hat{z}_{ij}^{(t)}},
\]

\[
\hat{V}_{j}^{(t+1)} = \frac{\sum_{i=1}^{n} \hat{z}_{ij}^{(t)} (X_i - \hat{M}_{j}^{(t+1)})^T (\hat{U}_{j}^{(t+1)})^{-1} (X_i - \hat{M}_{j}^{(t+1)})}{r \sum_{i=1}^{n} \hat{z}_{ij}^{(t)}}.
\]

Similar to the m.l.e. of the matrix normal distribution, \( \hat{U}_{j} \) and \( \hat{V}_{j} \) are obtained by updating the previously estimated parameters in the EM algorithm until a convergence criterion is satisfied.

### 2.2. Penalized Matrix Normal Mixture Model (PMNMM)

Let \( M_{g(k,l)} \) be the \((k, l)\)th element of the mean matrix of group \( g \), \( M_{g} \). If \( M_{g(k,l)} \) is the same for all groups, i.e., \( M_{1(k,l)} = \ldots = M_{G(k,l)} \), then the \((k, l)\)th element of the mean matrix cannot differentiate the clusters. In this case, when the data are standardized, the \((k, l)\)th element of the standardized mean matrix of each cluster will be 0, and is known to be the noise variable (non-informative) for clustering \([2]\). Therefore, penalizing on the elements of mean matrix of the matrix normal mixture was proposed by \([9]\) for identifying the underlying spatial structure and improving the signal-to-noise ratio (SNR) of the local field potentials (LFP) signals. However, in order to improve the clustering performance in high-dimension/low sample size data, the regularization of the precision matrix is inevitable because the number of parameters required to estimate the multiple covariance matrices becomes very large. In this paper, the method of \([9]\) is extended by further regularizing the precision matrix to cope better with the curse of dimensionality and reflect the proper conditional correlation structure for matrix data.

#### 2.2.1. Penalized Log-Likelihood Function of PMNMM

The penalized log-likelihood function is considered as follows:

\[
L_p(\Theta; \Lambda, X) = \sum_{i=1}^{n} \log \left( \sum_{j=1}^{G} \pi_{ij} f(X_i | \Theta_{ij}) \right) - \lambda_1 \sum_{j=1}^{G} |M_{j}| - \lambda_2 \sum_{j=1}^{G} |U_{j}^{-1}| - \lambda_3 \sum_{j=1}^{G} |V_{j}^{-1}|,
\]

where \(\lambda_1, \lambda_2, \lambda_3 > 0\) are the tuning parameters, and \(|A|\) is the L1 norm, which is the sum of all absolute values of elements of \( A \). Setting \(\lambda_2, \lambda_3 = 0\) is equivalent to the penalized
matrix normal mixture model with lasso regularization of the mean matrix proposed in [9]. Additionally, setting \( \lambda_1, \lambda_2, \lambda_3 = 0 \) is equivalent to the non-regularized matrix mixture model proposed in [5].

2.2.2. EM Algorithm of PMNMM

The EM algorithm is employed as in Section 2.1.3. The E-step is the same as Equation (4). In the M-step, the parameter estimates can be obtained by solving the constraint optimization problem as follows:

\[
\Theta^{(t+1)} = \arg\max_{M_t, U_t, & V_t > 0} \left\{ L(\Theta; X) - \lambda_1 \sum_{j=1}^{G} |M_j|_1 - \lambda_2 \sum_{j=1}^{G} |U_j|_1 - \lambda_3 \sum_{j=1}^{G} |V_j|_1 \right\},
\]

where \( U_t \) and \( V_t \) are symmetric and positive definite matrices. The estimate of \( M \) can be obtained by the subgradient approach as shown in Appendix B as follows:

\[
(M_{j}^{(t+1)})_{(k,l)} = \text{sign}(M_{j}^{(t)})_{(k,l)} \left( \frac{\lambda_1}{\sum_{i=1}^{n} \hat{\tau}_{ij}} U_{j}^{(t)} \right)_{1,p,q} V_{j}^{(t)} = \frac{\lambda_1}{\sum_{i=1}^{n} \hat{\tau}_{ij}} U_{j}^{(t)} \left( \hat{V}_{j}^{(t)} \right)_{1,p,q} V_{j}^{(t)} = \frac{\lambda_1}{\sum_{i=1}^{n} \hat{\tau}_{ij}} U_{j}^{(t)} \left( \hat{V}_{j}^{(t)} \right)_{1,p,q} V_{j}^{(t)}
\]

where \( M_{j}^{(t)} = \sum_{i=1}^{n} \frac{1}{\hat{\tau}_{ij}} X_{i}^{(t)} X_{i}^{(t)\top} \) is the m.l.e. of MNMM, \( |A|_{(k,l)} = \max(A_{(k,l)}), 0) \) is the matrix with all components of 1, and \( \text{sign}(A)_{(k,l)} \) is the function that extracts the sign of a real number, and \( \text{abs}(A) \) is the absolute value of \( A \). Then, we focus on the constraint optimization problem with respect to each \( U_t \) and \( V_t \). To maximize the log-likelihood with the constraint, the problem is redefined as the constraint optimization with respect to \( U_{j}^{-1} \) and \( V_{j}^{-1} \) as follows:

\[
\arg\max_{U_t > 0} \sum_{j=1}^{G} \left( q \sum_{i=1}^{n} \hat{\tau}_{ij} (U_{j}^{-1}) - \frac{q \sum_{i=1}^{n} \hat{\tau}_{ij}}{2} \text{tr}(S_{j}^{(U)} U_{j}^{-1}) \right) - \lambda_2 \sum_{j=1}^{G} |U_{j}|_1 \right\}
\]

with

\[
S_{j}^{(U)} = \frac{\sum_{i=1}^{n} \hat{\tau}_{ij} (X_{i} - \hat{M}_{j}) (X_{i} - \hat{M}_{j})^{\top}}{q \sum_{i=1}^{n} \hat{\tau}_{ij}}
\]

and

\[
\arg\max_{V_t > 0} \sum_{j=1}^{G} \left( p \sum_{i=1}^{n} \hat{\tau}_{ij} (V_{j}^{-1}) - \frac{p \sum_{i=1}^{n} \hat{\tau}_{ij}}{2} \text{tr}(S_{j}^{(V)} V_{j}^{-1}) \right) - \lambda_3 \sum_{j=1}^{G} |V_{j}|_1 \right\}
\]

with

\[
S_{j}^{(V)} = \frac{\sum_{i=1}^{n} \hat{\tau}_{ij} (X_{i} - \hat{M}_{j})^{\top}(\hat{U}_{j}^{-1}) (X_{i} - \hat{M}_{j})}{p \sum_{i=1}^{n} \hat{\tau}_{ij}}
\]

However, for Equations (6)–(9), the subgradient approach is not possible because the solution of the constraint optimization problem must satisfy the conditions that \( U_t \) and \( V_t \) are symmetric and positive definite matrices. So, the graphical lasso algorithm [12] is applied to maximize the following constraint optimization function with respect to positive definite matrix \( \Sigma \):

\[
\log|\Sigma^{-1}| - \text{tr}(S \Sigma^{-1}) - \lambda \Sigma^{-1}
\]

where \( S \) is the sample covariance matrix. Therefore, we can apply the algorithm by substituting \( S = S_{j}^{(U)} \), \( \Sigma = U_t \), \( \lambda = 2\lambda_2 / q \sum_{i=1}^{n} \hat{\tau}_{ij} \) and \( S = S_{j}^{(V)}, \Sigma = V_t, \lambda = 2\lambda_3 / p \sum_{i=1}^{n} \hat{\tau}_{ij} \) to estimate \( U_{j}^{-1} \) and \( V_{j}^{-1} \), respectively. The graphical lasso algorithm is implemented with R package glasso [12]. Then the m.l.e. of \( U_{j}^{-1} \) and \( V_{j}^{-1} \) are obtained by iterating E-step
and M-step until the penalized log-likelihood converges. To prevent the convergence of the parameter estimation from prematurely stopping, Aitken’s acceleration [19] is used. Aitken’s acceleration computes the acceleration coefficient at iteration \( t \) as follows:

\[
a^{(t)} = \frac{l^{(t)} - l^{(t-1)}}{l^{(t-1)} - l^{(t-2)}},
\]

where \( l^{(t)} \) is the penalized log-likelihood value at iteration \( t \). The accelerated maximum of the penalized log-likelihood at iteration \( t \) is as follows:

\[
\tilde{l}^{(t)} = l^{(t-1)} + \frac{l^{(t)} - l^{(t-1)}}{1 - a^{(t)}},
\]

The iteration stops when \( 0 \leq (\tilde{l}^{(t)} - l^{(t)}) / |l^{(t)}| \leq \epsilon \) for very small positive number \( \epsilon \). We set \( \epsilon = 0.001 \).

The iterative parameter estimation procedure needs the initial starting values. So the K-means clustering method is used to obtain the initial parameter values. For the fixed number of clusters \( G \), the vectorized data are divided into \( G \) clusters, using K-means. Then, based on a group label that is predicted, the m.l.e. are obtained for each cluster; we use those values as the initial value set for the EM algorithm. Additionally, in the process of estimating the parameters that maximize the objective function, the EM algorithm has to be run multiple times because local maximum values may exist.

### 2.3. Asymptotic Theory

The consistency of the penalized likelihood estimator for the mean of the mean-regularized matrix normal mixture model [9] under the following mild conditions was shown by the direct application of Theorem 1 in [20]. That is, the parameter space \( \Psi_{d_1,d_2} \) must be defined as follows:

\[
\Psi_{d_1,d_2} = \{ (\pi_1, \ldots, \pi_G; M_1, \ldots, M_G, V_1 \otimes U_1, \ldots, V_G \otimes U_G) \in \Psi_{d_1,d_2} : \frac{\sigma_i(U_h)}{\sigma_i(V_h)} = c_h \}
\]

for \( i = 1, \ldots, \min\{p,q\} \) and \( h = 1, \ldots, G \) (10)

to guarantee the identifiability of the matrix normal mixture model.

Since Theorem 1 in [20] is applicable to all lasso type penalized likelihood estimators, the penalized likelihood estimators of the proposed model are also consistent under the condition (10). We further establish the asymptotic distributions of the estimators, and prove their consistency in another way. We derive asymptotic distributions of the proposed estimators, using the techniques in [4]. The proof is given in Appendix C.

**Theorem 1.** Let \( X_1, \ldots, X_n \) be a random sample from a mixture matrix normal distribution. Let \( \hat{M}_j, \hat{U}_j^1, \) and \( \hat{V}_j^1 \) be the penalized maximum likelihood estimator of \( M_j, U_j^{-1}, \) and \( V_j^{-1} \), respectively, \( j = 1, \ldots, G \). Assume that the true parameter value \( M_j, U_j, V_j \in \Psi_{d_1,d_2} \), and \( \sqrt{n}\lambda_i \rightarrow \lambda_0 \geq 0 \) as \( n \rightarrow \infty, i = 1, 2, 3; \) then,

\[
\sqrt{n}(\hat{M}_j - M_j) \xrightarrow{d} \text{argmin}(V_1),
\]

where \( V_1(H) = -\text{tr}(U_j^{-1}HV_j^{-1}W_j^T) - \text{tr}(V_j^{-1}H^TU_j^{-1}W_1) + \text{tr}(V_j^{-1}H^TU_j^{-1}H) + \lambda_0 \sum_I \sum_m\{H(l,m)\text{sign}(\hat{M}_j(l,m))I(\hat{M}_j(l,m) \neq 0) + |H(l,m)|I(\hat{M}_j(l,m) = 0)\}, \)
in which $H$ is a $p \times q$ matrix, $H(i, m)$ is the $(i, m)$th element of $H$, $W_1$ is a random $p \times q$ matrix such that $\text{vec}(W_1) \sim N(0, \Lambda_1)$, and $\Lambda_1$ is such that the following holds:

$$\text{cov}(W_1(i, j), W_1(i', j')) = \text{cov}(X(i, j) - M_j(i, j), X(i', j') - M_j(i', j')).$$

(2) \[ \sqrt{n}(\hat{U}_j^* - U_j^{-1}) \xrightarrow{d} \arg\min_{L=L'} (V_2), \]

where

$$V_2(L) = \text{tr}(LU_jLU_j) + \text{tr}(LW_2) + \lambda_{20} \sum\limits_{l \neq m} \left\{ L(l, m) \text{sign}(\hat{U}_j^*(l, m)) I(\hat{U}_j^*(l, m) \neq 0) + |L(l, m)| I(\hat{U}_j^*(l, m) = 0) \right\},$$

in which $L$ is a $p \times p$ matrix, $L(l, m)$ is the $(l, m)$th element of $L$, $W_2$ is a random $p \times p$ matrix such that $\text{vec}(W_2) \sim N(0, \Lambda_2)$, and $\Lambda_2$ is such that the following holds:

$$\text{cov}(W_2(i, j), W_2(i', j')) = \text{cov}\left\{ (x_i^* - m_i^*)V_j^{-1}(x_i^* - m_i^*)^T / q, (x_{i'}^* - m_{i'}^*)V_j^{-1}(x_{i'}^* - m_{i'}^*)^T / q \right\},$$

where $x_i^*$ and $m_i^*$ is the $i$th row of the matrix $X$ and $M_j$, respectively.

(3) \[ \sqrt{n}(\hat{V}_j^* - V_j^{-1}) \xrightarrow{d} \arg\min_{R=R'} (V_3), \]

where

$$V_3(R) = \text{tr}(RV_jRV_j) + \text{tr}(RW_3) + \lambda_{30} \sum\limits_{l \neq m} \left\{ R(l, m) \text{sign}(\hat{V}_j^*(l, m)) I(\hat{V}_j^*(l, m) \neq 0) + |R(l, m)| I(\hat{V}_j^*(l, m) = 0) \right\},$$

in which $R$ is a $q \times q$ matrix, $R(l, m)$ is the $(l, m)$th element of $R$, $W_3$ is a random $q \times q$ matrix such that $\text{vec}(W_3) \sim N(0, \Lambda_3)$, and $\Lambda_3$ is such that the following holds:

$$\text{cov}(W_3(i, j), W_3(i', j')) = \text{cov}\left\{ (x_i - m_i)^T U_j^{-1}(x_i - m_i) / p, (x_{i'} - m_{i'})^T U_j^{-1}(x_{i'} - m_{i'}) / p \right\},$$

where $x_i$ and $m_i$ is the $i$th column of the matrix $X$ and $M_j$, respectively.

**Corollary 1.** Assume that $M_j, U_j, V_j \in \Psi_{d_1,d_2}$ and $\sqrt{n}\lambda_i \to \lambda_{i0} \geq 0$ as $n \to \infty$, $i = 1, 2, 3$, then $\hat{M}_j, \hat{U}_j^*$, and $\hat{V}_j^*$ are consistent.

**Proof.** Since $\sqrt{n}(\hat{M}_j - M_j)$ converges in a distribution, $\sqrt{n}(\hat{M}_j - M_j) = O_p(1)$ and $|\hat{M}_j - M_{j1}| = O_p(1/\sqrt{n}) = o_p(1)$ as $n \to \infty$. Therefore, the proposed estimator $\hat{M}_j$ is $\sqrt{n}$ consistent. The proof for $\hat{U}_j^*$ and $\hat{V}_j^*$ are same as for $\hat{M}_j$. \qed

### 2.4. Model Selection

In the penalized mixture model, the selection of the optimal hyperparameters is very important. The proposed penalized matrix normal mixture model has four hyperparameters ($G, \lambda_1, \lambda_2, \lambda_3$). A grid search method is used to select the optimal set of $(G, \lambda_1, \lambda_2, \lambda_3)$ with the largest penalized log-likelihood. However, even if the grid search method is applied, a model fitted with a fixed dataset is often not reliable due to the overfitting problem. Generally, the k-fold cross-validation (CV) method is used to solve these problems. In k-fold CV, the data $X = \{X_1, X_2, \ldots, X_n\}$ are split into $K$ mutually exclusive subsets denoted by $Y_1, Y_2, \ldots, Y_k$. Then, one of the $K$ subsets is selected as the test data, the model with the remaining data is applied, and the penalized log-likelihood value for the test data is calculated with the estimated parameters. After using each of the subsets as test data
in turn, the final score is obtained by averaging the K penalized log-likelihood calculated values as follows:

\[ L_p(K) = \frac{1}{K} \sum_{k=1}^{K} L_p(Y_k | Y_{-k}) \]

where \( Y_{-k} \) is the dataset excluding \( Y_k \). For example, in 4-fold cross-validation, the mean of the penalized log-likelihood is obtained as shown in Figure 1.

Figure 1. 4-fold cross-validation.

Therefore, using the k-fold cross-validation method, we can select the optimal parameter set with the largest mean of the penalized log-likelihood among different combinations of grid parameter values.

3. Experiments

3.1. Simulation Studies

In Scenarios 1 and 2, we generate 20 \( \times \) 20 data from two matrix normal distributions \( N_{400}(\text{vec}(M_1), V_1 \otimes U_1) \), \( N_{400}(\text{vec}(M_2), V_2 \otimes U_2) \), each with a different type of mean structure. One has a rectangle shape and the other has a cross shape, as in Figure 2.

Figure 2. Mean structure of each group.

We fix \( G = 2 \) in the simulation study, and the ranges of the tuning parameter \( \lambda_1, \lambda_2, \lambda_3 \) are set to be \( \lambda_1 = 20, 10, 5, 1, 0.5, 0.1, \lambda_2 = 1, 0.1, 0.01, 0.001, \lambda_3 = 1, 0.1, 0.01, 0.001 \), and the optimal \((\lambda_1, \lambda_2, \lambda_3)\) set is determined with the 4-fold CV method. The sample size \( n \) is set to be 100, 200, 300, and 1000. To compare the performance for different degrees of sparsity, we consider different precision structures for Scenario 1 and Scenario 2, banding and AR(1) model, respectively:

Scenario 1: Banding precision structure

\[
(V_1 \otimes U_1) = \Sigma_1^{-1} = \begin{cases} 
1 & \text{if } i = j \\
0.2 & \text{if } |i - j| = 1, \ 1 \leq i, j \leq 400. \\
0 & \text{otherwise} 
\end{cases}
\]

\[
(V_2 \otimes U_2) = \Sigma_2^{-1} = \begin{cases} 
1 & \text{if } i = j \\
0.2 & \text{if } |i - j| = 1 \\
0.3 & \text{if } |i - j| = 2, \ 1 \leq i, j \leq 400. \\
0 & \text{otherwise} 
\end{cases}
\]
Scenario 2: AR(1) precision structure

\((V_1 \otimes U_1) = \Sigma_1^{-1} = 0.5^{i-j}, \quad 1 \leq i, j \leq 400.\)

\((V_2 \otimes U_2) = \Sigma_2^{-1} = 0.4^{i-j}, \quad 1 \leq i, j \leq 400.\)

The proposed model is compared with the conventional methods: K-means, spectral clustering, and the model of [9]. The criteria for performance are ACC and adjusted rand index (ARI) [21]. ACC shows the closeness of a predicted value to a standard or known value, and ARI is a clustering evaluation method that indicates perfect clustering if it is 0 and random clustering if it is 0. Additionally, the two norms are considered to show the consistency of the parameter estimates of the model: the averaged spectral norm and Frobenius norm of the difference between the estimated precision matrix and the truth.

\[
SL = \frac{1}{G} \sum_{j=1}^{G} \|\hat{\Sigma}_j^{-1} - \Sigma_j^{-1}\|_s,
\]

\[
FL = \frac{1}{G} \sum_{j=1}^{G} \|\hat{\Sigma}_j^{-1} - \Sigma_j^{-1}\|_F
= \frac{1}{G} \sum_{j=1}^{G} \sqrt{\sum_{i} \sum_{j} (\hat{\Sigma}_j^{-1}(i, j) - \Sigma_j^{-1}(i, j))^2},
\]

where \(\|A\|_s\) is the largest singular value of matrix \(A\), and \(\|A\|_F\) is the Frobenius norm.

Since the clustering results depend on the generated data, each scenario is run 50 times. ACC, ARI, SL, and FL are calculated each time, and their average results are reported in Tables 1–4, respectively. Figures 3–6 show the violin plots of the ACC and ARI for each method with a box plot and underlying distribution by sample size in Scenario 1 and 2.

Table 1. Scenario 1: Performance comparison for PMNMM and conventional clustering methods (The values in parentheses are the standard deviations of the 50 runs’ results).

| n   | PMNMM | Method of [9] | K-Means | Spectral |
|-----|-------|---------------|---------|----------|
|     | ACC   | ARI           | ACC     | ARI      | ACC     | ARI     | ACC     | ARI     |
| 100 | 0.995 | 0.98          | 0.915   | 0.706    | 0.754   | 0.288   | 0.838   | 0.489   |
|     | (0.01) | (0.041)       | (0.07)  | (0.213)  | (0.097) | (0.184) | (0.097) | (0.198) |
| 200 | 0.999 | 0.999         | 0.997   | 0.999    | 0.875   | 0.572   | 0.936   | 0.763   |
|     | (0.002) | (0.002)       | (0.005) | (0.018)  | (0.06)  | (0.16)  | (0.021) | (0.071) |
| 300 | 1     | 1             | 1       | 1        | 0.932   | 0.747   | 0.961   | 0.849   |
|     | (0.0005) | (0.002)       | (0.001) | (0.005)  | (0.026) | (0.088) | (0.014) | (0.051) |
| 1000| 1     | 1             | 1       | 1        | 0.978   | 0.916   | 0.988   | 0.954   |
|     | (0.0002) | (0.0008)      | (0.0002) | (0.0008) | (0.005) | (0.019) | (0.003) | (0.013) |

Table 2. Scenario 1: Mean measured errors of the parameter estimates for the selected optimal parameter set (The values in parentheses are the standard deviations of the 50 runs’ results).

| n   | Optimal Parameter Set | FL (Mean) | FL (Precision) | SL (Mean) | SL (Precision) | FL (Mean) | FL (Precision) |
|-----|-----------------------|-----------|----------------|-----------|----------------|-----------|----------------|
| 100 | \(\lambda_1 = 20\)  | \(\lambda_2 = 0.001\) | \(\lambda_3 = 0.01\) | 2.9       | 0.926          | 5.34      | (0.078)        |
|     |                       |           |                |           | (0.071)        | (0.168)  |
| 200 | \(\lambda_1 = 0.1\)  | \(\lambda_2 = 0.01\) | \(\lambda_3 = 0.1\) | 2.19      | 0.597          | 3.9       | (0.048)        |
|     |                       |           |                |           | (0.045)        | (0.088)  |
| 300 | \(\lambda_1 = 0.1\)  | \(\lambda_2 = 0.001\) | \(\lambda_3 = 0.001\) | 1.79      | 0.465          | 3.39      | (0.048)        |
|     |                       |           |                |           | (0.028)        | (0.048)  |
| 1000| \(\lambda_1 = 0.1\)  | \(\lambda_2 = 0.001\) | \(\lambda_3 = 0.001\) | 0.983     | 0.291          | 2.6       | (0.026)        |
|     |                       |           |                |           | (0.01)         | (0.025)  |
Table 3. Scenario 2: Performance comparison for PMNMM and conventional clustering methods (The values in parentheses are the standard deviations of the 50 runs’ results).

| n   | PMNMM  | Method of [9] | K-means | Spectral |
|-----|--------|---------------|---------|----------|
|     | ACC    | ARI           | ACC     | ARI      | ACC    | ARI    | ACC    | ARI    |
| 100 | 0.665  | 0.135         | 0.671   | 0.137    | 0.636  | 0.105  | 0.674  | 0.155  |
|     | (0.094)| (0.146)       | (0.086) | (0.122)  | (0.1)  | (0.132)| (0.104)| (0.155) |
| 200 | 0.847  | 0.548         | 0.753   | 0.286    | 0.678  | 0.155  | 0.811  | 0.4    |
|     | (0.132)| (0.334)       | (0.093) | (0.182)  | (0.091) (0.143) | (0.062) | (0.147) |
| 300 | 0.991  | 0.964         | 0.909   | 0.705    | 0.743  | 0.269  | 0.881  | 0.583  |
|     | (0.006)| (0.024)       | (0.097) | (0.182)  | (0.095) (0.143) | (0.062) | (0.147) |
| 1000| 0.999  | 0.994         | 0.998   | 0.993    | 0.968  | 0.876  | 0.971  | 0.888  |
|     | (0.001)| (0.005)       | (0.004) | (0.004)  | (0.009) (0.032) | (0.006) | (0.022) |

Table 4. Scenario 2: Mean measured errors of the parameter estimates for the selected optimal parameter set (The values in parentheses are the standard deviations of the 50 runs’ results).

| n   | Optimal Parameter Set | FL       | SL       | FL       |
|-----|-----------------------|----------|----------|----------|
|     | λ₁, λ₂, λ₃            | (Mean)   | (Precision) | (Precision) |
| 100 | 0.1, 0.001, 0.1       | 4.51     | 1.23     | 6.77     |
|     |                       | (0.253)  | (0.155)  | (0.372)  |
| 200 | 0.1, 0.001, 0.001     | 2.93     | 0.797    | 5.34     |
|     |                       | (0.384)  | (0.061)  | (0.24)   |
| 300 | 0.1, 0.001, 0.001     | 2.02     | 0.621    | 4.71     |
|     |                       | (0.062)  | (0.047)  | (0.054)  |
| 1000| 0.1, 0.001, 0.001     | 1.11     | 0.422    | 4.11     |
|     |                       | (0.026)  | (0.013)  | (0.027)  |

Figure 3. Violin plot of ACC in Scenario 1.
Figure 4. Violin plot of ARI in Scenario 1.

Figure 5. Violin plot of ACC in Scenario 2.
Figure 6. Violin plot of ARI in Scenario 2.

In Scenario 1, Figures 3 and 4 and Table 1 show that the proposed model has the highest ACC (0.995) and ARI (0.98) for sample size 100; the method of [9] follows next. In addition, as n increases, the performance of PMNMM and the method of [9] become similar. To see if there is any significant difference among the means of the ACC and ARI, the Kruskal–Wallis test [22] is applied. The p-value of the test for significant difference among the means of ACC and ARI is 0, so we can confirm that the ACC and ARI means of the models differ under the significance level $\alpha = 0.05$. Then, to compare pairwise difference, the Dunn test [23] is applied. The null hypothesis of the Dunn test is that there is no difference between the two groups. As a result, PMNMM and the method of [9] is better than both K-means and the spectral method ($p$-value = 0), and the spectral method is better than K-means ($p$-value = 0) in all sample sizes. For a small sample case ($n = 100$), the proposed model shows better performance ($p$-value = 0) because there are many zero-valued elements of the precision matrix in the outside of the band, which are estimated to be relatively better than other methods. For the other sample size, $p$-values of the Dunn test between the ACC of PMNMM and the method of [9] is 0.9436, 1, and 1, respectively, and 0.8309, 1, 1 for ARI, respectively. Because the null hypothesis cannot be rejected, the performance of the two models is considered to be the same. As shown in Figures 3 and 4, the ACC and ARI values of the K-means method, spectral method and the method of [9] are more spread than those of PMNMM. This implies that the proposed method produces more stable clustering results.

Table 2 shows the mean measured errors between estimated parameters and true parameters for 50 runs in Scenario 1. As the sample size increases, the FL value of the estimated mean matrix decreases, and similarly, the SL and FL values of the estimated precision matrix decrease. To see if there is any significant difference among values for each sample size, the Kruskal–Wallis test and the Dunn test are applied. With the Kruskal–Wallis test, the $p$-values for the means of FL (mean), SL (precision), and FL (precision) are 0, so the means of measured errors for three norms differ among different sample sizes under the significance level $\alpha = 0.05$. Next, with the Dunn test, we can confirm that the means of each error decrease as the sample size increases, as $p$-values are smaller than the significance
level $\alpha = 0.05$ for two different increasing sample sizes. This implies that the proposed method has consistency.

In Scenario 2, similar to the previous process, we can check if there is any significant difference among the means of the ACC and ARI. However, under the AR(1) precision structure, the results are slightly different. For the sample size $n = 100$, the $p$-values of the Kruskal–Wallis test are 0.19 and 0.17 for ACC and ARI, respectively, so we cannot reject the null hypothesis that the means of the ACC and ARI for all models are the same. For the other sample sizes, the $p$-value of the Kruskal–Wallis test is 0, so the Dunn test is applied. For $n = 200$, the $p$-value of the Dunn test between PMNMM and spectral is 0.3911, so the null hypothesis cannot be rejected. On the contrary, the $p$-value of the Dunn test between PMNMM and the method of [9] is 0, so PMNMM is better than the method of [9]. The sparse precision matrix is not shown clearly for the data with $n = 100$ sample size in Scenario 2, and this hinders PMNMM from exploring sparsity in the precision matrix, showing similar performance as the conventional method. However, for $n = 300$, the $p$-values of the Dunn test between any pair of methods are 0. We can confirm that PMNMM is better than the K-means and spectral methods, as well as the method of [9]. For $n = 1000$, PMNMM and the method of [9] have the same performance ($p$-values = 1, 1 for ACC and ARI, respectively), and are better than the other two methods, but the performance of the other two methods is the same ($p$-values = 0.3433, 0.3294).

Table 4 shows the mean measured errors between estimated parameters and true parameters in scenario 2. Table 4 shows similar results as Table 2, and the Kruskal–Wallis test and the Dunn test are applied in the same way as before. As a result, for the Kruskal–Wallis test, the $p$-values for the test of significant difference among the means of each measured error are 0, and for the Dunn test, the $p$-values for all pairwise comparisons are 0, so all null hypotheses are rejected under the significance level $\alpha = 0.05$. This implies that the proposed method has consistency.

In the simulation studies, the two precision structures are considered: banding and AR(1). The experimental results of Scenario 1 showed that the performance of the proposed method is better than the other methods in all sample size cases; we can confirm the consistency of the proposed method through the measured errors with the increasing sample size. The penalized matrix normal mixture model in the banding precision matrix showed good performance, even with a very small sample size. However, in the AR(1) precision structure of Scenario 2, a larger number of samples was needed to achieve good performance, and the value of the measured error was higher when compared with Scenario 1. This implies that the proposed model works better for data with a more sparse precision structure.

3.2. Real Data Studies

- **Datasets**

  We evaluated the proposed model on the CIFAR 10 and Fashion-MNIST datasets:

  - **CIFAR 10**

    The CIFAR 10 dataset ([https://www.cs.toronto.edu/~kriz/cifar.html](https://www.cs.toronto.edu/~kriz/cifar.html), accessed on 21 September 2021) contains well-known image data. There are 60,000 images of $32 \times 32 \times 3$ size with 10 types—airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck—and each type has 6000 images, respectively. The dataset is divided into a training set with 50,000 images and a test set with 10,000 images.

  - **Fashion-MNIST**

    The Fashion-MNIST dataset ([https://github.com/zalandoresearch/fashion-mnist](https://github.com/zalandoresearch/fashion-mnist), accessed on 21 September 2021) is a grayscale image dataset, designed to replace the original MNIST dataset. There are 70,000 images of $28 \times 28$ size with 10 types—T-shirt/top, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag, ankle boot—and each type has 7000 images, respectively. The dataset is divided into a training set with 60,000 images and a test set with 10,000 images.
3.2.1. CIFAR 10 Dataset

In the experiment, two types of images are selected: frog and ship (Figure 7). For the performance comparison of the proposed method, the training images are transformed to grayscale, and 500 images are randomly selected from the training set for each type. In addition, the K-means and spectral methods, as well as the method of [9] are the conventional methods to be compared with PMNMM; the number of clusters \( G \) is set to 2; and the ranges of the tuning parameter \( \lambda_1, \lambda_2, \lambda_3 \) are set to \( \lambda_1 = 1, 0.5, 0.1, \lambda_2 = 1, 0.5, 0.1, 0.05, 0.01, \lambda_3 = 1, 0.5, 0.1, 0.05, 0.01 \). Using the training set, the optimal set of \( (\lambda_1, \lambda_2, \lambda_3) \) selected through the 4-fold CV method is \((1, 0.01, 0.01)\), and the performance of the proposed model is evaluated on the testing set. A total of 500 images are randomly selected from the test set, and the models are fitted to obtain ACC and ARI, respectively. This process is repeated 50 times to calculate the average ACC and ARI of each model, and the results are shown in Table 5. Then, the Kruskal–Wallis test and the Dunn test are applied to see if there is any significant difference between the methods. As a result, the \( p \)-value of the Kruskal–Wallis test is 0, the ACC and ARI of the PMNMM are higher than those of K-means and spectral method with the Dunn test (\( p \)-value = 0), but PMNMM and the method of [9] have the same performance. The null hypothesis of the Dunn test between the K-means and spectral methods is not rejected (\( p \)-value = 0.075, 0.085 for ACC and ARI, respectively) under the significance level \( \alpha = 0.05 \).

![Figure 7. Two types of images in CIFAR 10 dataset: frog and ship.](image)

Table 5. Performance comparison of PMNMM with the conventional methods for CIFAR 10 dataset (The values in parentheses are the standard deviations of the 50 runs’ results).

| Method       | ACC   | ARI   |
|--------------|-------|-------|
| PMNMM        | 0.7418 (0.014) | 0.2335 (0.026) |
| [9]’s method | 0.7418 (0.014) | 0.2335 (0.026) |
| K-means      | 0.6654 (0.015) | 0.1088 (0.019) |
| Spectral     | 0.6751 (0.017) | 0.1222 (0.023) |

To compare PMNMM with [9]’s method further, a more extreme low sample situation is set up by randomly selecting 10 images from the training set for each type. The results are shown in Table 6.
Table 6. Performance comparison of PMNMM and [9]’s method for CIFAR 10 dataset (The values in parentheses are the standard deviations of the 50 runs’ results).

| Method        | ACC       | ARI       |
|---------------|-----------|-----------|
| PMNMM         | 0.7206    | 0.1936    |
|               | (0.042)   | (0.0753)  |
| [9]’s method  | 0.7204    | 0.1934    |
|               | (0.043)   | (0.0743)  |

Tables 5 and 6 show that the performance does not improve significantly even if the sample size increases because the two groups are difficult to be distinguished. There is also little performance difference between the two methods because the weight of the penalty imposed on the precision matrix is small.

3.2.2. Fashion Dataset

We consider applying the methods to the fashion data, which can be distinguished more clearly. Similar to the CIFAR 10 data application, two types of images are selected—shirt and sandal—and 500 images are randomly selected from the training set for each type (Figure 8). The ranges of the tuning parameter $\lambda_1, \lambda_2, \lambda_3$ are set to $\lambda_1 = 1, 0.5, 0.1$, $\lambda_2 = 5, 3, 1, 0.5, 0.1$, $\lambda_3 = 5, 3, 1, 0.5, 0.1$. Using the training set, the optimal set of $(\lambda_1, \lambda_2, \lambda_3)$ selected through the 4-fold CV method is $(0.1, 0.1, 0.1)$. The results for ACC and ARI are shown in Table 7. It is confirmed that PMNMM and the method of [9] have almost the same performance for the sample size $n = 1000$. Then, the Kruskal–Wallis and Dunn tests are applied; the $p$-value of the Kruskal–Wallis test is 0; the ACC and ARI of the PMNMM are higher than those of K-means and spectral methods with the Dunn test ($p$-value = 0); and the null hypothesis of the Dunn test between the K-means and spectral methods is not rejected ($p$-value = 1) under the significance level $\alpha = 0.05$.

Figure 8. Two types of images in CIFAR 10 dataset: frog and ship.

Table 7. Performance comparison of PMNMM with the conventional methods for the fashion dataset (The values in parentheses are the standard deviations of the 50 runs’ results).

| Method        | ACC       | ARI       |
|---------------|-----------|-----------|
| PMNMM         | 0.9504    | 0.8113    |
|               | (0.007)   | (0.024)   |
| [9]’s method  | 0.9500    | 0.8095    |
|               | (0.014)   | (0.026)   |
| K-means       | 0.8828    | 0.5865    |
|               | (0.015)   | (0.046)   |
| Spectral      | 0.8831    | 0.5870    |
|               | (0.013)   | (0.040)   |
To compare PMNMM with the method of [9] further, a more extreme low sample situation is set up by randomly selecting 10 images from the training set for each type. The results are shown in Table 8.

Table 8. Performance comparison of PMNMM and the method of [9] for the fashion dataset (The values in parentheses are the standard deviations of the 50 runs’ results).

| Method       | ACC       | ARI       |
|--------------|-----------|-----------|
| PMNMM        | 0.9342    | 0.7541    |
|              | (0.025)   | (0.086)   |
| [9]’s method | 0.6754    | 0.1207    |
|              | (0.026)   | (0.037)   |

Table 8 shows that the proposed method works better than the method of [9] that imposes a penalty only on the mean matrix for extremely small sample size data.

4. Conclusions

We have proposed a new penalized mixture model with the penalties imposed both on the mean matrix and precision matrix of the matrix normal mixture model. The conventional Gaussian mixture model for matrix data was challenging because it has to estimate one big covariance structure; therefore, the calculation speed becomes very slow as the dimension increases. On the contrary, the proposed method is a parsimonious model since it reduces a large number of parameters by regularizing the row and column variance matrices of the matrix normal distribution, especially imposing a lasso penalty on the precision matrix. Therefore, the proposed method can reflect the sparse conditional correlation structure and work even with high-dimensional data with a small sample size. The asymptotic distributions of the parameter estimators were derived, and their consistency was proved. We have shown the superior performance of the proposed model by comparing it with three conventional methods (method of [9], K-means method, and spectral method) through both synthetic and real data experiments. A performance comparison was done with ACC and ARI, and PMNMM was significantly better than the K-means and spectral methods by a nonparametric test. In the simulation studies, we evaluated the clustering performance for two different degree of sparsity of precision matrix. For the banding precision structure, the ACC performance of the proposed method reached 0.995 and 0.999, with the small sample size, n = 100 and 200, respectively, which are higher than the other methods. For the AR(1) precision structure, it is similar to that of [9] with n = 100 and higher than those of the other methods with n = 200, 300, and 1000. In conclusion, when the true precision matrix is sparse, that is, the variables have strong conditionally independent structure, the proposed model is well fitted to a small sample size data. For the real datasets, the ACC performance of the proposed method reaches 0.74 and 0.95 for clustering two types of CIFAR 10 and Fashion-MNIST images, respectively, which are higher than or equal to those of the other methods.

In the proposed model, calculation speed is an important factor, as mentioned above. Thus, the graphical lasso algorithm [12] was applied. A faster algorithm to estimate a large precision matrix would be beneficial for the proposed model to improve the calculation speed. In addition, to cluster a 3D image or higher dimensional data, new penalized mixture models for the tensor structure need to be developed. A penalized matrix non-Gaussian (such as skew-normal, skew-t) mixture model is worth developing for clustering more general types of matrix data.

Author Contributions: Conceptualization, J.H. and J.B.; methodology, J.H. and J.B.; software, J.H.; validation, J.H.; formal analysis, J.H.; investigation, J.H.; resources, J.H.; writing—original draft preparation, J.H.; writing—review and editing, J.B.; visualization, J.H.; supervision, J.B.; project administration, J.B. All authors have read and agreed to the published version of the manuscript.
**Funding:** This research of the first author was supported by BK21 FOUR (Fostering Outstanding Universities for Research, NO.5120200913674) funded by the Ministry of Education (MOE, Korea) and National Research Foundation of Korea (NRF) and was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2018S1A6A3A04042721). The research of the second author was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A1B07049729).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on https://www.cs.toronto.edu/~kriz/cifar.html, accessed on 21 September 2021 and https://github.com/zalandoresearch/fashion-mnist, accessed on 21 September 2021. The R code for the proposed method is available on request from the first author.

**Acknowledgments:** The authors thank two reviewers for their valuable comments for the improvement of the manuscript.

**Conflicts of Interest:** The authors declare no conflict of interest.

**Appendix A. A Comparative Overview Table**

|                          | K-Means | Spectral | Gaussian Mixture with Regularized Mean (9) | Mixture of Matrix Normal with Regularized Mean/Covariance (Proposed Method) |
|--------------------------|---------|----------|--------------------------------------------|--------------------------------------------------------------------------|
| Data type                | Vector  | Vector/Matrix | Vector                                   | Matrix                           | Matrix                                       |
| Clustering type          | Centroid-based | Similarity-based | Model-based            | Model-based                  | Model-based                                  |
|                         |         |          |    |                                       | |                                    |
| Affine transformation invariance | X | X | O | O | O |
| Number of parameters     | NA      | NA       | Gpq(pq + 1)/2 + Gpq + G − 1 | Gp(p + 1)/2 + Gq(q + 1)/2 + Gpq + G − 1 − r | Gp(p + 1)/2 + Gq(q + 1)/2 + Gpq + G − 1 − r * |
| (p × q data, number of clusters: G) |         |          |    |                                       | |                                    |
| Modeling ability         | X       | X        | X | O | O |
| for covariance structure of matrix data |         |          |    |                                       | |                                    |

$r$: the number of mean parameters values that are shrunken to zero (0 ≤ r ≤ Gpq); $r^*$: the number of mean and precision parameters values that are shrunken to zero; (r ≤ $r^*$ ≤ Gp(p + 1)/2 + Gq(q + 1)/2 + Gpq); NA: not applicable.

**Appendix B**

**Proof of Equation (5).**

\[
\frac{\partial L_p(\Theta; \Lambda)}{\partial M_j} = \frac{\partial}{\partial M_j} \left[ \sum_{i=1}^{n} \sigma_{ij} \left\{ -\frac{1}{2} tr(V_j^{-1}(X_i - M_j)^T U_j^{-1}(X_i - M_j)) \right\} - \lambda_1 |M_j|_1 \right] = 0,
\]

using the formula \( \frac{\partial}{\partial X} tr(AX^T BX) = BXA^T + B^TXA \) (KB Petersen, 2012 [24]), we obtain the following:
\[
\frac{\partial}{\partial M_j} \left[ \sum_{i=1}^{n} \tau_{ij} \left\{ -\frac{1}{2} \text{tr}(V_j^{-1}(X_i - M_j)^T U_j^{-1}(X_i - M_j)) \right\} - \lambda_1 |M_j|_1 \right]
\]

\[
= \sum_{i=1}^{n} \tau_{ij} \left\{ U_j^{-1}(X_i - M_j)V_j^{-1} \right\} - \lambda_1 \text{sign}(M_j) = 0.
\]

Some algebraic manipulations are applied as follows:

\[
\frac{\sum_{i=1}^{n} \tau_{ij} \left\{ U_j^{-1}(X_i - M_j)V_j^{-1} \right\}}{\sum_{i=1}^{n} \tau_{ij}} - \frac{\lambda_1}{\sum_{i=1}^{n} \tau_{ij}} \text{sign}(M_j) = 0.
\]

\[
\Rightarrow \quad U_j^{-1} \frac{\sum_{i=1}^{n} \tau_{ij}X_i}{\sum_{i=1}^{n} \tau_{ij}} V_j^{-1} - U_j^{-1}M_jV_j^{-1} - \frac{\lambda_1}{\sum_{i=1}^{n} \tau_{ij}} \text{sign}(M_j) = 0.
\]

\[
\Rightarrow \quad \frac{\sum_{i=1}^{n} \tau_{ij}X_i}{\sum_{i=1}^{n} \tau_{ij}} - M_j - \frac{\lambda_1}{\sum_{i=1}^{n} \tau_{ij}} U_j \text{sign}(M_j)V_j = 0.
\]

Because \(\frac{\sum_{i=1}^{n} \tau_{ij}X_i}{\sum_{i=1}^{n} \tau_{ij}}\) is the estimate of \(M_j\) for the non-penalty model, the following equation is derived:

\[
\tilde{M}_j = M_j - \frac{\lambda_1}{\sum_{i=1}^{n} \tau_{ij}} U_j \text{sign}(\tilde{M}_j)V_j
\]

\[
= \text{sign}(\tilde{M}_j) \left[ |\tilde{M}_j| - \frac{\lambda_1}{\sum_{i=1}^{n} \tau_{ij}} U_j 1_{p \times q} V_j \right]. \quad (A1)
\]

\[
\square
\]

\textbf{Appendix C}

\textbf{Proof of Theorem 1.} Let \(U_j^*\) and \(V_j^*\) be the inverse matrix of \(U_j\) and \(V_j\), respectively. Then, the penalized Q function can be rewritten in the M-step as follows.

\[
Q_p = \sum_{j=1}^{G} \sum_{i=1}^{n} \left[ -\frac{pq}{2} \log(2\pi) + \frac{p}{2} \log|V_j^*| + \frac{q}{2} \log|U_j^*| - \frac{1}{2} \text{tr} \left\{ V_j^*(X_i - M_j)^T U_j^*(X_i - M_j) \right\} \right]
\]

\[
- \lambda_1 \sum_{j=1}^{G} |M_j|_1 - \lambda_2 \sum_{j=1}^{G} |U_j^*|_1 - \lambda_3 \sum_{j=1}^{G} |V_j^*|_1.
\]

We took a similar approach as that which [4] used to derive the asymptotic distribution of the precision matrix estimator in the Gaussian graphical model. \(\square\)

\textbf{Appendix C.1. The Limiting Dist. of \(\tilde{M}_j\)}

Let \(\tilde{M}_j\) be the matrix that minimizes the following:

\[
\frac{1}{2} \sum_{i=1}^{n} \tau_{ij} \text{tr} \left\{ V_j^*(X_i - M_j)^T U_j^*(X_i - M_j) \right\} + \lambda_1 |\tilde{M}_j|_1, \quad \text{or}
\]

\[
\frac{1}{2} \sum_{i=1}^{n} \tau_{ij} \text{tr} \left\{ V_j^*(X_i - M_j)^T U_j^*(X_i - M_j) \right\} / \sum_{i=1}^{n} \tau_{ij} + \lambda_1 |\tilde{M}_j|_1,
\]

where \(\lambda_1^* = 2\lambda_1 / \sum_{i=1}^{n} \tau_{ij}\).

Define \(V_{1n}(H)\) as the following:
\[ V_{1n}(H) = \sum_{i=1}^{n} \tau_{ij} tr\{V_i^{*}(X_i - M_j - H/\sqrt{n})^{T}U_j^{*}(X_i - M_j - H/\sqrt{n})\} / \sum_{i=1}^{n} \tau_{ij} \]

\[ + \lambda_1^{*} \sum_{i=1}^{p} \sum_{m=1}^{q} |\tilde{M}_j(l, m) + H(l, m)/\sqrt{n}| \]

\[ - \sum_{i=1}^{n} \tau_{ij} tr\{V_i^{*}(X_i - M_j)^{T}U_j^{*}(X_i - M_j)\} / \sum_{i=1}^{n} \tau_{ij} + \lambda_1^{*} \sum_{i=1}^{p} \sum_{m=1}^{q} |\tilde{M}_j(l, m)|. \]

Note the following:

\[ \sum_{i=1}^{n} \tau_{ij} tr\{V_i^{*}(X_i - M_j - H/\sqrt{n})^{T}U_j^{*}(X_i - M_j - H/\sqrt{n})\} / \sum_{i=1}^{n} \tau_{ij} \]

\[ - \sum_{i=1}^{n} \tau_{ij} tr\{V_i^{*}(X_i - M_j)^{T}U_j^{*}(X_i - M_j)\} / \sum_{i=1}^{n} \tau_{ij} \]

\[ = - tr\left\{ V_i^{*}\left( \sum_{i=1}^{n} \tau_{ij}\left( X_i - M_j \right)\right)^{T}U_j^{*}(H/\sqrt{n}) \right\} - tr\left\{ V_i^{*}(H/\sqrt{n})^{T}U_j^{*}\left( \sum_{i=1}^{n} \tau_{ij}\left( X_i - M_j \right)\right) \right\} \]

\[ + tr(V_i^{*}H^{T}U_j^{*}H/\sqrt{n}) \]

\[ = - tr\left\{ U_j^{*}H^{T}(\tilde{M}_j - M_j)^{T}\right\} / \sqrt{n} - tr\left\{ V_j^{*}H^{T}U_j^{*}(\tilde{M}_j - M_j)\right\} / \sqrt{n} \]

\[ + tr(V_j^{*}H^{T}U_j^{*}H)/\sqrt{n}, \]

where \( \tilde{M}_j = \sum_{i=1}^{n} \tau_{ij}X_i / \sum_{i=1}^{n} \tau_{ij}, \) which is the m.l.e. of \( M_j \) for the non-penalized matrix normal mixture model.

Together with the fact that

\[ \lambda_1^{*} \sum_{i=1}^{p} \sum_{m=1}^{q} \left( |\tilde{M}_j(l, m) + H(l, m)/\sqrt{n}| - |\tilde{M}_j(l, m)| \right) \]

\[ = \lambda_1^{*} \sum_{i=1}^{p} \sum_{m=1}^{q} \left\{ H(l, m)\text{sign}(\tilde{M}_j(l, m)) \text{I}(\tilde{M}_j(l, m) \neq 0) + |H(l, m)| \text{I}(\tilde{M}_j(l, m) = 0) \right\}, \]

\( nV_{1n}(H) \) can be rewritten as follows:

\[ - tr(U_j^{-1} HV_j^{-1} W_{1n}) - tr(V_j^{-1} H^{T}U_j^{-1} W_{1n}) + tr(V_j^{-1} H^{T} U_j^{-1} H) \]

\[ + \sqrt{n}\lambda_1^{*} \sum_{i=1}^{p} \sum_{m=1}^{q} \left\{ H(l, m)\text{sign}(\tilde{M}_j(l, m)) \text{I}(\tilde{M}_j(l, m) \neq 0) + |H(l, m)| \text{I}(\tilde{M}_j(l, m) = 0) \right\}, \]

where \( W_{1n} = \sqrt{n}(\tilde{M}_j - M_j) \) and \( \text{vec}(W_{1n}) \overset{d}{\to} N(0, \Lambda_1). \) Therefore, \( nV_{1n}(H) \overset{d}{\to} V(H). \)

Since both \( V_1(H) \) and \( nV_{1n}(H) \) are convex and \( V_1(H) \) has a unique minimum, we have the following:

\[ \text{argmin } nV_{1n}(H) = \sqrt{n}(\tilde{M}_j - M_j) \overset{d}{\to} \text{argmin } V_1(H) \]

\[ \text{(A2)} \]

Appendix C.2. The Limiting Dist. of \( \tilde{U}_j^{*} \)

Let \( \tilde{U}_j^{*} \) be the positive definite matrix that minimizes the following:

\[ - \frac{q}{2} \sum_{i=1}^{n} \tau_{ij} \log |\tilde{U}_j^{*}| + \frac{q}{2} \sum_{i=1}^{n} \tau_{ij} tr(\tilde{U}_j^{*}S_j^{(U)}) + \lambda_2 |\tilde{U}_j^{*}|_1, \]
where \( S_j^{(U)} = \sum_{i=1}^p \tau_{ij} (X_i - M_j) V_j^T (X_i - M_j)^T \), which is the m.l.e. of \( U_j \) for the non-penalized matrix normal mixture model, or minimizes the following:

\[
- \log |\tilde{U}_j^*| + \text{tr}(\tilde{U}_j^* S_j^{(U)}) + \lambda_2^* |\tilde{U}_j^*|_1,
\]

where \( \lambda_2^* = 2 \lambda_2 / (q \sum_{i=1}^q \tau_{ij}) \).

Define \( V_{2n}(L) \) as follows:

\[
V_{2n}(L) = - \log |\tilde{U}_j^* + L / \sqrt{n}| + \text{tr}\left\{ (\tilde{U}_j^* + L / \sqrt{n}) S_j^{(U)} \right\} + \lambda_2^* \sum_{l \neq m} |\tilde{U}_j^*(l, m) + L(l, m) / \sqrt{n}|
+ \log |\tilde{U}_j^*| - \text{tr}(\tilde{U}_j^* S_j^{(U)}) + \lambda_2^* \sum_{l \neq m} |\tilde{U}_j^*(l, m)|.
\]

Note the following:

\[
\log |\tilde{U}_j^* + L / \sqrt{n}| - \log |\tilde{U}_j^*| = \log |1 + \frac{LU_j^{1/2}LU_j^{1/2}}{\sqrt{n}}| = \sum_{i=1}^p \log \left\{ 1 + \sigma_i(LU_j^{1/2}LU_j^{1/2}/\sqrt{n}) \right\},
\]

where \( \sigma_i(\cdot) \) denotes the \( i \)th-largest eigenvalue of a matrix. Since

\[
\log \left\{ 1 + \sigma_i(LU_j^{1/2}LU_j^{1/2}/\sqrt{n}) \right\} = \frac{\sigma_i(LU_j^{1/2}LU_j^{1/2})}{\sqrt{n}} - \frac{\sigma_i^2(LU_j^{1/2}LU_j^{1/2})}{n} + o(\frac{1}{n}),
\]

we conclude the following:

\[
\log |\tilde{U}_j^* + L / \sqrt{n}| - \log |\tilde{U}_j^*| = \sum_{i=1}^p \frac{\sigma_i(LU_j^{1/2}LU_j^{1/2})}{\sqrt{n}} - \frac{\text{tr}(LU_j^{1/2}LU_j^{1/2})}{n} + o(\frac{1}{n})
= \frac{\text{tr}(LU_j^{1/2}LU_j^{1/2})}{\sqrt{n}} - \frac{\text{tr}(LU_j^{1/2}LU_j^{1/2})}{n} + o(\frac{1}{n})
= \frac{\text{tr}(LU_j)}{\sqrt{n}} - \frac{\text{tr}(LU_j)}{n} + o(\frac{1}{n}).
\]

On the other hand,

\[
\text{tr}\left\{ (\tilde{U}_j^* + L / \sqrt{n}) S_j^{(U)} \right\} - \text{tr}(\tilde{U}_j^* S_j^{(U)}) = \text{tr}\left\{ \frac{LS_j^{(U)}}{\sqrt{n}} \right\}
= \frac{\text{tr}(LU_j)}{\sqrt{n}} + \frac{\text{tr}\left\{ L(S_j^{(U)} - U_j) \right\}}{\sqrt{n}}.
\]

Together with the fact that

\[
\lambda_2^* \sum_{l \neq m} (|\tilde{U}_j^*(l, m) + L(l, m) / \sqrt{n}| - |\tilde{U}_j^*(l, m)|)
= \frac{\lambda_2^*}{\sqrt{n}} \sum_{l \neq m} \left\{ L(l, m) \text{sign}(\tilde{U}_j^*(l, m)) I(\tilde{U}_j^*(l, m) \neq 0) + |L(l, m)| I(\tilde{U}_j^*(l, m) = 0) \right\},
\]
\( nV_{2n}(L) \) can be rewritten as follows:

\[
tr(LU_jLU_j) + tr(LW_{2n}) + \sqrt{n} \lambda^*_j \sum_{l \neq m} \left\{ \frac{L(l, m) \text{sign}(U_j^*(l, m)) I(U_j^*(l, m) \neq 0) + |L(l, m)| I(U_j^*(l, m) = 0) \right\} + o(1),
\]

where \( W_{2n} = \sqrt{n}(S_j^{(U)} - U_j) \) and \( \text{vec}(W_{2n}) \xrightarrow{d} N(0, \Lambda_2) \) because \( S_j^{(U)} \) is the m.l.e. of \( U_j \) for non-penalized matrix normal mixture model. Since both \( V_2(L) \) and \( nV_{2n}(L) \) are convex and \( V_2(L) \) has a unique minimum, the following holds:

\[
\text{argmin } nV_{2n}(L) = \sqrt{n}(U_j^* - U_j) \xrightarrow{d} \text{argmin } V_2(L) \quad (A3)
\]

**Appendix C.3. The Limiting Dist. of \( V_j^* \)**

Let \( \bar{V}_j^* \) be the positive definite matrix that minimizes the following:

\[
- \frac{p \sum_{i=1}^n \tau_{ij}}{2} \log |\bar{V}_j^*| + \frac{p \sum_{i=1}^n \tau_{ij}}{2} tr(\bar{V}_j^* S_j^{(V)}) + \lambda_3 \bar{V}_j^* |_{1,}
\]

where \( S_j^{(V)} = \frac{\sum_{i=1}^n (x_i - M_j)^T U_j^*(x_i - M_j)}{p \sum_{i=1}^n \tau_{ij}} \), or minimizes the following:

\[
- \log |\bar{V}_j^*| + tr(\bar{V}_j^* S_j^{(V)}) + \lambda_3^* |\bar{V}_j^* |_{1,}
\]

where \( \lambda_3^* = 2\lambda_3 / (p \sum_{i=1}^n \tau_{ij}) \).

Define \( V_{3n}(R) \) as follows:

\[
V_{3n}(R) = - \log |V_j^*| + \frac{R}{\sqrt{n}} \log |\bar{V}_j^*| + tr \left\{ (V_j + \frac{R}{\sqrt{n}}) S_j^{(V)} \right\} + \lambda_3^* \sum_{l \neq m} |V_j^*(l, m) + \frac{R(l, m)}{\sqrt{n}}|.
\]

\[
+ \log |\bar{V}_j^*| - tr(\bar{V}_j^* S_j^{(V)}) + \lambda_3^* \sum_{l \neq m} |\bar{V}_j^*(l, m)|
\]

Note the following:

\[
log |\bar{V}_j^*| + \frac{R}{\sqrt{n}} - log |\bar{V}_j^*| = log |I + \frac{V_j^{1/2} RV_j^{1/2}}{\sqrt{n}}| = \sum_{i=1}^q \log \left\{ 1 + \sigma_i(V_j^{1/2} RV_j^{1/2} / \sqrt{n}) \right\}.
\]

Since

\[
\log \left\{ 1 + \sigma_i(V_j^{1/2} RV_j^{1/2} / \sqrt{n}) \right\} = \frac{\sigma_i(V_j^{1/2} RV_j^{1/2} / \sqrt{n})}{\sqrt{n}} - \frac{\sigma_i^2(V_j^{1/2} RV_j^{1/2} / \sqrt{n})}{n} + o(\frac{1}{n}),
\]

we conclude the following:

\[
log |\bar{V}_j^*| + \frac{R}{\sqrt{n}} - log |\bar{V}_j^*| = \sum_{i=1}^q \frac{\sigma_i(V_j^{1/2} RV_j^{1/2} / \sqrt{n})}{\sqrt{n}} - \frac{tr(V_j^{1/2} RV_j^{1/2} / \sqrt{n})}{n} + o(\frac{1}{n})
\]

\[
= \frac{tr(V_j^{1/2} RV_j^{1/2} / \sqrt{n})}{n} - \frac{tr(V_j^{1/2} RV_j^{1/2})}{n} + o(\frac{1}{n})
\]

\[
= tr(RV_j) - tr(RV_j) / n + o(\frac{1}{n}).
\]
On the other hand,
\[
\text{tr}\left\{ (\tilde{V}_j^* + \frac{R}{\sqrt{n}}) S_j^{(V)} \right\} - \text{tr}(\tilde{V}_j^* S_j^{(V)}) = \frac{\text{tr}(RS_j^{(V)})}{\sqrt{n}} + \frac{\text{tr}\left\{ R(S_j^{(V)} - V_j) \right\}}{\sqrt{n}}.
\]
Together with the fact that
\[
\lambda_3^* \sum_{l \neq m} |(\tilde{V}_j^*(l, m) + \frac{R(l, m)}{\sqrt{n}}) - |\tilde{V}_j^*(l, m)|
\]
\[
= \lambda_3^* \sum_{l \neq m} \left\{ R(l, m) \text{sign}(\tilde{V}_j^*(l, m) I(\tilde{V}_j^*(l, m) \neq 0) + |R(l, m)| I(\tilde{V}_j^*(l, m) = 0) \right\},
\]
nV_{3n}(R) can be rewritten as follows:
\[
\text{tr}(RV_j^*V_j) + \text{tr}(RW_{3n})
\]
\[
+ \sqrt{n} \lambda_3^* \sum_{l \neq m} \left\{ R(l, m) \text{sign}(\tilde{V}_j^*(l, m) I(\tilde{V}_j^*(l, m) \neq 0) + |R(l, m)| I(\tilde{V}_j^*(l, m) = 0) \right\} + o(1),
\]
where \( W_{3n} = \sqrt{n}(S_j^{(V)} - V_j) \) and \( \text{vec}(W_{3n}) \overset{d}{\to} N(0, \Lambda_3) \) because \( S_j^{(V)} \) is the m.l.e. of \( V_j \) for non-penalized matrix normal mixture model. Since both \( V_3(R) \) and \( nV_{3n}(R) \) are convex and \( V_3(R) \) has a unique minimum, we have the following:
\[
\arg\min nV_{3n}(R) = \sqrt{n}(V_j^* - V_j) \overset{d}{\to} \arg\min V_3(R) \tag{A4}
\]

References
1. Pan, W.; Shen, X. Penalized Model-Based Clustering with Application to Variable Selection. J. Mach. Learn. Res. 2007, 8, 1145–1164.
2. Zhou, H.; Pan, W.; Xiaotong, S. Penalized model-based clustering with unconstrained covariance matrices. Electron. J. Stat. 2009, 3, 1473–1496.
3. Ruan, L.; Yuan, M.; Zou, H. Regularized Parameter Estimation in High-Dimensional Gaussian Mixture Models. Neural Comput. 2011, 23, 1605–1622.
4. Yuan, M.; Lin, Y. Model selection and estimation in the Gaussian graphical model. Biometrika 2007, 94, 19–35.
5. Viroli, C. Finite mixtures of matrix normal distributions for classifying three-way data. Stat. Comput. 2011, 21, 511–522.
6. Gallaugher, M.; McNicholas, P. A Mixture of Matrix Variate Skew-t Distributions. arXiv 2017, arXiv:1703.08882.
7. Melnykov, V.; Zhu, X. On model-based clustering of skewed matrix data. J. Multivar. Anal. 2018, 167, 181–194.
8. Tomarchio, S.; Punzo, A.; Bagnato, L. Two new matrix-variate distributions with application in model-based clustering. Comput. Stat. Data Anal. 2020, 152, 107050.
9. Gao, X.; Shen, W.; Zhang, L.; Hu, J.; Fortin, N.; Frostig, R.; Ombao, H. Regularized matrix data clustering and its application to image analysis. Biometrics 2020, 10, https://doi.org/10.1111/biom.13354.
10. Dempster, A.P. Covariance selection. Biometrics 1972, 28, 157–175.
11. Laritgue, T. Mixture of Gaussian Graphical Models with Constraints. Ph.D. Thesis, Institut Polytechnique de Paris, Palaiseau, France, 2020.
12. Friedman, J.; Hastie, T.; Tibshirani, R. Sparse inverse covariance estimation with the graphical lasso. Biostatistics 2007, 9, 432–441.
13. Lloyd, S. Least squares quantization in PCM. IEEE Trans. Inf. Theory 2006, 28, 129–137.
14. Ng, A.; Jordan, M.; Weiss, Y. On Spectral Clustering: Analysis and an algorithm. In Proceedings of the Conference on Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 9–14 December 2002; pp. 849–856.
15. Duttileau, P. The MLE algorithm for the matrix normal distribution. J. Stat. Comput. Simul. 1999, 64, 105–123.
16. Srivastava, M.; Rosen, T.; Rosen, D. Models with a Kronecker product covariance structure: Estimation and testing. Mathematical Methods of Statistics. Math. Methods Stat. 2008, 17, 357–370.
17. Dawid, A. Some matrix-variate distribution theory: Notational considerations and a Bayesian application. Biometrika 1981, 68, 265–274.
18. Dempster, A.; Laird, N.; Rubin, D. Maximum Likelihood From Incomplete Data Via The EM algorithm. *J. R. Stat. Soc. Ser. B (Methodological)* **2008**, *39*, 1–38.

19. Aitken, A. XXV.—On Bernoulli’s Numerical Solution of Algebraic Equations. *Proc. R. Soc. Edinb.* **1926**, *46*, 289–305.

20. Fan, J.; Li, R. Variable Selection via Nonconcave Penalized Likelihood and Its Oracle Properties. *J. Am. Stat. Assoc.* **2001**, *96*, 1348–1360.

21. Hubert, L.; Arabie, P. Comparing Partitions. *J. Classif.* **1985**, *2*, 193–218.

22. Kruskal, W.; Wallis, W. Errata: Use of Ranks in One-Criterion Variance Analysis. *J. Am. Stat. Assoc.* **1952**, *47*, 583–621.

23. Dunn, O. Multiple comparison using RANK sums. *Technometrics* **1964**, *6*, 241–252.

24. Petersen, K.B.; Pedersen, M.S. The Matrix Cookbook. 2012. Available online: [https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf](https://www.math.uwaterloo.ca/~hwolkowi/matrixcookbook.pdf) (accessed on 21 September 2021).