BAYESIAN SPARSE FACTOR ANALYSIS OF HIGH DIMENSIONAL GENE EXPRESSION DATA

A Thesis
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Jingjun Zhao

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Statistics

April 2019

Fargo, North Dakota

North Dakota State University
Graduate School

Title

BAYESIAN SPARSE FACTOR ANALYSIS OF HIGH DIMENSIONAL GENE EXPRESSION DATA

By

Jingjun Zhao

The Supervisory Committee certifies that this ***disquisition*** complies with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Gang Shen
Chair

Dr. Rhonda Magel

Dr. Juan Li

Approved:

May 6, 2019
Date

Dr. Rhonda Magel
Department Chair

ABSTRACT

This work closely studied fundamental techniques of Bayesian sparse Factor Analysis model - constrained Least Square regression, Bayesian Lasso regression, and some popular sparsity-inducing priors. In Appendix A, we introduced each of the fundamental techniques in a coherent manner and provided detailed proof for important formulas and definitions. We consider provided introduction and detailed proof, which are very helpful in learning Bayesian sparse Factor Analysis, as a contribution of this work.

We also systematically studied a computationally tractable biclustering approach in identifying co-regulated genes, *BicMix*, by proving all point estimates of the parameters and by running the method on both simulated data sets and a real high-dimensional gene expression data set. Missed derivation of all point estimates in *BicMix* has been provided for better understanding variational expectation maximization (VEM) algorithm. The performance of the method for identifying true biclusters has been analyzed using the experimental results.

ACKNOWLEDGEMENTS

I would like to thank my committee members: Dr. Shen Gang, Dr. Rhonda Magel, and Dr. Li Juan for their continued support and guideline.

I deeply thank my parents, Shengwu Zhao and Shuyan Xiang for their unconditional support. The family of my old sister have been generous with their love and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDIX FIGURES

1. INTRODUCTION

Data clustering techniques have been widely applied in high dimensional gene expression data analysis for different purposes, such as motif identification, functional annotation, and tissue classification. These data are generally organized as a matrix, where each row represents a gene, each column corresponds to an experimental condition, and each cell stands for the expression level of a gene under a specific condition (Griffin and Brown, 2010).

Among different clustering techniques for identifying groups of co-regulated genes in gene expression data, i.e., identifying subsets of genes with similar behaviors under subsets of experimental conditions, biclustering approach became very popular. It's a powerful data mining technique that allows clustering of rows and columns simultaneously in a matrix-format data set. This technique was first applied to gene expression data in 2000, aiming to identify co-expressed genes under a subset of all the conditions/samples. Since then, many biclustering algorithms have been developed and applied in identifying co-regulated genes.

Alternatively, latent factor models are often used to identify groups of co-regulated genes in gene expression data (West M., 2003). In particular, latent factor models decompose a matrix $Y \in \mathcal{R}^{p\times n}$ of p genes and n samples into the product of two matrices, $\Lambda \in \mathcal{R}^{p\times K}$, the factor loadings matrix, and $\mathrm{X} \in \mathcal{R}^{K\times n}$, the latent factor matrix, for K latent factors, and assuming independent Gaussian noise (Gao et. al, 2016). Latent factor models assume that the total variation within the gene-expression data matrix can be partitioned into covariation among genes and variation specific to genes. This implies that a set of genes with correlated gene expression levels will contribute substantially to (have a substantial loading on) a single factor, because this co-variability will contribute to the overall variability in the matrix.

Gao et. al (2016) developed a probabilistic biclustering method, *BiMix,* to infer subsets of co-regulated genes whose covariation may be observed in only a subset of the samples. It's a

computationally tractable method based on a Bayesian sparse latent factor model. Instead of imposing Laplace prior on loading matrix as in FABIA, *BicMix* adopted a different sparsity-inducing prior - three parameter Beta ($\mathcal{TPB}$) (Armagan et al, 2011) - to model the variance of the loading matrix. $\mathcal{TPB}$ prior is a generalized sparsity-inducing prior and allows *BicMix* have flexible shrinkage capability on both the loading matrix and factor matrix at three different levels: element-specific shrinkage, factor-specific shrinkage, and global shrinkage.

In this work, we closely studied Bayesian sparse Factor Analysis model and its related fundamental techniques. Proof for some of the formulas and definitions in these techniques has been provided. It will be helpful to better understand Bayesian sparse latent Factor Analysis, especially for new learners. We also systematically studied *BicMix* method by proving all model parameter estimates and running the method on simulated and high-dimensional gene expression data.

2. LITERATURE REVIEW

Biclustering algorithms that are capable to simultaneously cluster rows and columns of a data matrix have been successfully applied to gene expression data to discover co-regulated genes over a subset of conditions. Here a bicluster is a submatrix that consists of a subset of rows and a subset of columns in a matrix, and contains homogenous patterns. Some biclustering algorithms use hierarchical clustering to group together similar samples and features (Ben et al., 2003; Murali and Kasif, 2003). Li et al., (2009) proposed a biclustering method to build up biclusters by iteratively grouping features in a greedy way, i.e., identifying all genes that have correlated expression levels with a selected gene—and then removing samples that do not support that grouping. Factor analysis for bicluster acquisition (FABIA), a newer probabilistic model-based biclustering method (Hochreiter et al., 2010) adopted a Bayesian sparse factor analysis model to decompose a gene expression matrix into two sparse matrices. Sparsity-inducing priors, such as the Laplace prior, are imposed on elements of both the loading and the factor matrices to induce zero-valued elements.

Factor analysis is a statistical modeling technique that seeks to explain correlation among observed, correlated variables in terms of smaller number of unobserved or latent (hidden) causal factors. Such models are called latent factor models. This technique has been used in diverse areas to extract useful low dimensional features from high dimensional data (Yuna et al, 2010, Joseph et al, 2010). In areas of genetics, latent factor models, especially sparse latent factor models, have been used to identify groups of co-regulated genes in high-dimensional gene expression data (Carvalho et al., 2008; Engelhardt and Stephens, 2010). Because of concerns with identifiability, it's assumed that gene expression levels for each gene is a linear combination of latent factors and that the random noise is approximately normal. Therefore, each sample is modeled as being drawn from a multivariate normal distribution with

a diagonal covariance matrix across genes, where the mean parameter is a linear combination of latent factors with a normal prior, and the variance term is estimated for each feature separately (Gao et al., 2013).

In the Bayesian context, a number of sparsity inducing priors have been proposed. Popular used priors include Laplace (also called Lasso/double-exponential), Cauchy, Strawderman-Berger, normal-Feffreys, normal-exponential-gamma, Horseshoe (Carvalho et al., 2010), and three-parameter-beta (Armagan et al, 2011). All of these priors can be represented by a scale mixture of Gaussians that makes identifying relationships among different sparsity inducing priors become easier. The proof that a sparsity-inducing prior can be equally represented by a scale mixture of Gaussians is provided in Appendix A 3.1.

A horseshoe prior (Carvalho et al., 2010) is for shrinkage in the presence of sparsity in a data set, and it's able to adapt to different sparsity patterns while simultaneously avoiding the over-shrinkage of large coefficients, that means it has heavy tails. In a simple situation where $(y|\beta) \sim N(\beta, \sigma^2 I)$ and where β is believed to be sparse. The horseshoe prior assumes that each β_i is conditionally independent with density $\pi_{HS}(\beta_i|\tau)$, where π_{HS}can be represented as a scale mixture of Gaussians:

$$(\beta_i|\lambda_i, \tau) \sim N\left(0, {\lambda_i}^2 \tau^2\right) \tag{2.1}$$

$$\lambda_i \sim C^{+(0,1)} \tag{2.2}$$

where $C^{+(0,1)}$ is a half-Cauchy distribution for the standard deviation λ_i. Here, λ_i is local shrinkage parameters and τ is a global shrinkage parameter. The prior has a shrinkage coefficient, k_i , with distribution

$$P_k(k_i\ ;\ \tau\ ) = \frac{\tau}{\pi} \frac{1}{1-(1-\tau^2)k_i} (1-k_i)^{-\frac{1}{2}} {k_i}^{-\frac{1}{2}} \tag{2.3}$$

Variable k_i represents the amount of weight that posterior mean for β_i places on 0 once y has been observed. The posterior mean $E[\beta_i|y_i]$ is given as:

$$T_\tau(y) = E[\beta_i|y_i] = (1 - E[(k_i|y_i)])y_i. \tag{2.4}$$

The posterior of $(k_i|y_i)$ is given as:

$$P(k_i|y_i) \propto exp(-k_i {y_i}^2/2) \times \frac{\tau}{\pi} \frac{1}{1-(1-\tau^2)k_i} (1 - k_i)^{-\frac{1}{2}}, \tag{2.5}$$

and the detailed derivation is provided in Appendix A 3.2.

The shrinkage effect in $T_\tau(y)$ can be adjusted by value of τ. For example, decreasing τ will skew the prior distribution on k_i towards zero, corresponding to more mass near zero. That means large τ values are more likely to be sampled, which results in a higher prior probability of shrinking the observations toward zero. The figure 2.1 graphically shows distribution of k_i with different τ, decreasing τ results in a higher prior probability of shrinking the observations toward zero.

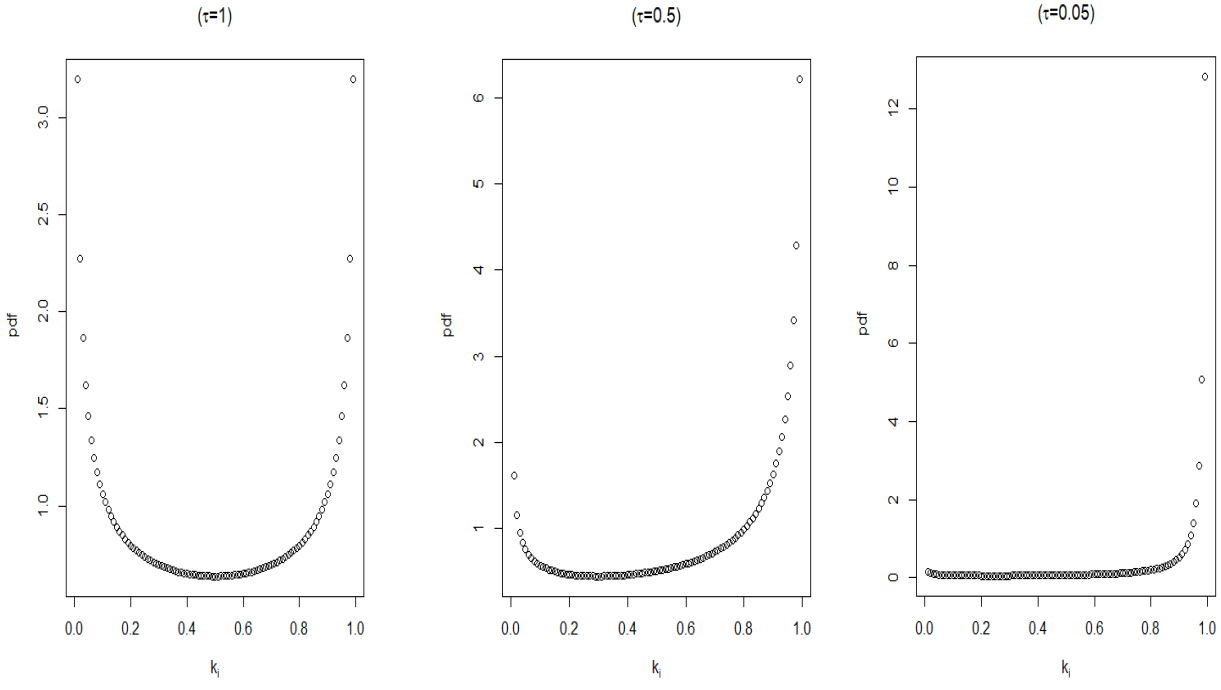

Figure 2.1. The effect of decreasing τ (1, 0.5, 0.05) on the priors of k_i.

Detailed introduction to Horseshoe prior and detailed derivation of the equations above can be found in Appendix 3.2.

Among proposed sparsity inducing priors, none that specifically address the context of a high dimensional latent space. Armagan et al. (2011) developed a three parameter Beta ($\mathcal{TPB}$) prior, a generalization of the beta distribution to form a flexible class of scale mixtures of normal with very appealing behavior. Given specific settings of hyperparameters, TPB can recapitulate sparsity inducing priors with appropriate modeling assumptions. *BicMix*, a probabilistic biclustering method, used $\mathcal{TPB}$ distribution to induce sparsity in both the factors (samples) and the loadings (genes). Detailed introduction to $\mathcal{TPB}$ is provided in Appendix 3.3.

3. METHODOLOGY

To identify biclusters in gene expression data, Gao et. al (2016) developed a Bayesian sparse Factor Analysis model, *BicMix*. The method used a simple factor analysis model along with a general sparsity-inducing prior, which was imposed on both the loading matrix and factor matrix at three different levels - element-specific shrinkage, factor-specific shrinkage, and global shrinkage. Since orthogonality assumption in standard Factor Analysis is violated in gene expression data where correlated sources of variation may impact similar subsets of genes, *BicMix* doesn't require orthogonality across the factors or loadings. Without orthogonality constraints, it is possible that many of these components explain similar variation in the observations, which is expected in gene expression data.

To thoroughly understand model *BicMix*, comprehending essential modeling techniques including constrained Lease Square regression, Bayesian Lasso regression, sparsity-inducing priors, and Variational Expectation Maximation (VEM) is indispensable. Therefore, we complemented work of *BicMix* by providing proper introduction of these required techniques in a rational order, along with providing missed proof or detailing existed proof for some of important formulas and definitions appearing in the techniques. The complement is provided in Appendix A. Model *BicMix* adopted VEM method for parameter estimation. To make point estimates of the parameters (over 80) presenting in the original paper clear and easy to understand, we provided detailed derivation of the estimates in Appendix B. The derivation facilitates practical study of VEM method.

In this chapter, we first introduced similarity between a model for p biclusters and a Factor Analysis model for p factors, then we introduced *BicMix* model in detail.

3.1. Biclustering Model

A bicluster is defined as a pair of a row (gene) set and a column (sample) set for which the rows are similar to each other on the columns and vice versa, and such a linear dependency on subsets of rows and columns can be represented as an outer product λz^T of two sparse vectors λ and z, shown in Figure 3.1 below (Hochreiter et al., 2010). The non-zero entries in the vectors are adjacent to each other are for visualization purpose only.

Therefore, the overall model for p biclusters and additive noise is

$$Y = \sum_{i=1}^{p} \lambda_i z_i^T + \Upsilon = \Lambda Z + \Upsilon \tag{3.1}$$

where $\Upsilon \in \mathbb{R}^{n \times l}$ is additive noise; $\lambda_i \in \mathbb{R}^n$ and $z_i \in \mathbb{R}^{n \times l}$ are the sparse prototype vector and the sparse vector of factors of the i^{th} bicluster, respectively. The second formulation above holds if $\Lambda \in \mathbb{R}^{n \times p}$ is the sparse prototype matrix containing the prototype vectors λ_i as columns and $Z \in \mathbb{R}^{p \times l}$ is the sparse factor matrix containing the transposed factors z_i^T as rows. Note that Eq. (3.1) formulates biclustering as sparse matrix factorization (or decomposition).

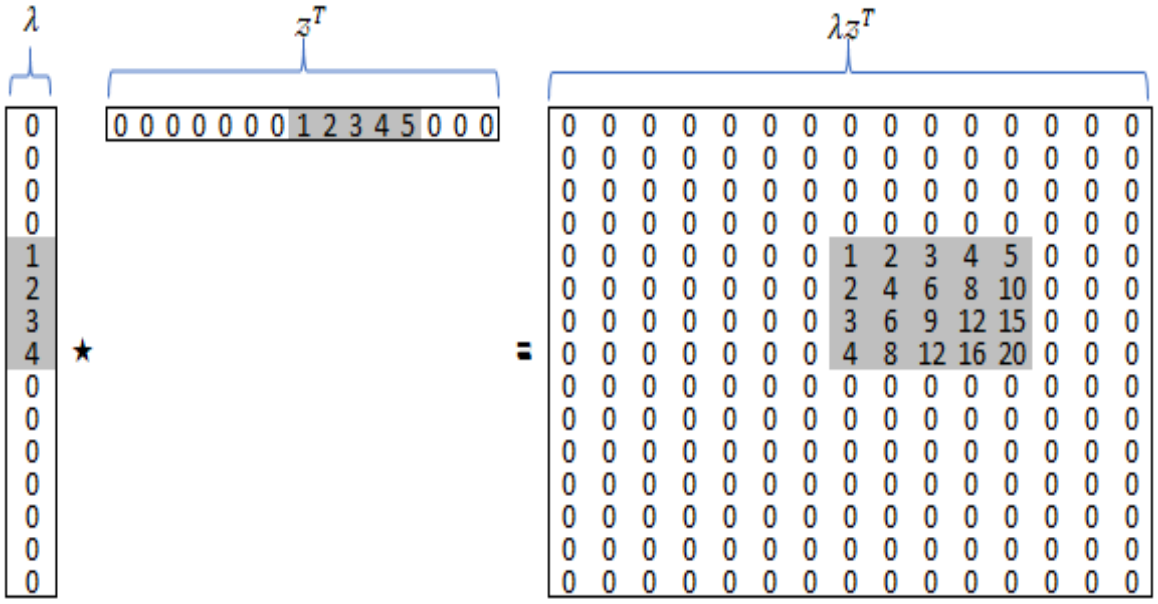

Figure 3.1. The outer product λz^T of two sparse vectors results in a matrix with a bicluster

According to Eq. (3.1), the j^{th} sample x_j, i.e. the j^{th} column of Y, is

$$y_j = \sum_{i=1}^{p} \lambda_i z_{i,j} + \epsilon_j = \Lambda \tilde{z_j} + \epsilon_j \tag{3.2}$$

where ϵ_j is the j^{th} column of the noise matrix Υ and $\tilde{z_j} = (z_{1,j}, \dots, z_{p,j})^T$ denotes the j^{th}column of the matrix Z. Recall that $z_i^T = (z_{i,1}, \dots, z_{i,l})$ is the vector of values that constitutes the $\boldsymbol{i^{th}}$ bicluster (one value per sample), while $\tilde{z_j}$ is the vector of values that contribute to the $\boldsymbol{j^{th}}$ sample (one value per bicluster).

The formulation in Eq. (3.2) facilitates a generative interpretation by a factor analysis model with p factors (Everitt, 1984)

$$Y = \sum_{i=1}^{p} \lambda_i \tilde{z_\iota} + \epsilon = \Lambda \tilde{Z} + \epsilon \tag{3.3}$$

where Y is the observation, Λ is the loading matrix, $\tilde{z_\iota}$is the value of the i^{th} factor, $\tilde{Z} = (\widetilde{z_1}, \dots, \widetilde{z_p})^T$is the vector of factors and $\epsilon \in \mathbb{R}^n$ is the additive noise. Standard factor analysis assumes: the noise is independent of $\tilde{Z}$, $\tilde{Z} \sim \mathcal{N}(0, \mathrm{I})$ and $\epsilon \sim \mathcal{N}(0, \Psi)$, $\Psi \in \mathbb{R}^{n \times n}$is diagonal. The parameter Λ explains the depend (common) and Ψ the independent variance in the observations Y. That the covariance matrix for $\tilde{Z}$ is the unit matrix (I) means that the biclusters should not be correlated.

3.2. BicMix

The defined Bayesian sparse factor analysis model for *BicMix* is shown below

$$Y = \Lambda \mathrm{X} + \epsilon, \tag{3.4}$$

where $Y \in \mathcal{R}^{p \times n}$ is the matrix of observed variables; $\Lambda \in \mathcal{R}^{p \times K}$ is the loading matrix, $\mathrm{X} \in \mathcal{R}^{K \times n}$ is the factor matrix; and $\epsilon \in \mathcal{R}^{p \times n}$ is the residual error matrix for p genes and n samples. Assumption for the noise is $\epsilon_{.,i} \sim \mathcal{N}(0, \Psi)$, where $\Psi = diag(\psi_1, \dots, \psi_p)$. It's easy to get

$$E[Y] = \Lambda \mathrm{X} = \sum_{i=1}^{K} \lambda_i x_i' \tag{3.5}$$

where K is the number of latent factors, λ_i is a loading vector, and x_i is a factor vector. The number of K must be initialized, and it should be set as an overestimate of the number of latent factors. In chapter Numerical Experiments, K was initialized based on the size of analyzed data. For *BicMix* results, components that were classified as sparse have each element threshold at 10^{-10}, because adopted parameter estimation methods converged to values near, but not exactly, zero. This model removes factors that are unsupported in the data through a sparse inducing prior. By imposing significant sparsity on the loading matrix, rotational invariance in the basic factor model (Kaiser, 1958) for the most part can be eliminated.

BicMix adopted Three Parameter Beta ($\mathcal{TPB}$) prior (Armagan et al, 2011) to model the variance of Λ. $\mathcal{TPB}$ is a generalization of the beta distribution to form a flexible class of scale mixtures of normal with very appealing behavior. The three-parameter distribution has the form

$$f(x;, a, b, \phi) = \frac{\Gamma(\mathrm{a}+\mathrm{b})}{\Gamma(\mathrm{a})\Gamma(\mathrm{b})} \phi^b x^{b-1} (1-x)^{a-1} \{1 + (\phi - 1)x\}^{-(a+b)} \tag{3.6}$$

for $0 < x < 1$, $a > 0$, $b > 0$, and $\phi > 0$. It's denoted by $\mathcal{TPB}(a, b, \phi)$. Given specific settings of hyperparameters, $\mathcal{TPB}$ can recapitulate sparsity inducing priors with appropriate modeling assumptions and computational, e.g., $\mathcal{TPB}(a = 0.5, b = 0.5, \phi = 1)$ is equivalent to Horseshoe prior, a popular used spare-inducing prior. Next, we describe the sparsity-inducing structure for Λ and X. The hierarchical structure for Λ is written as

$$\Lambda_{\mathrm{j,k}} \sim \mathcal{N}(0, \frac{1}{\varphi_{j,k}} - 1) \tag{3.7}$$

$$\varphi_{\mathrm{j,k}} \sim \mathcal{TPB}\left(a, \mathrm{b}, \frac{1}{\xi_{\mathrm{k}}} - 1\right) \tag{3.8}$$

$$\xi_k \sim \mathcal{TPB}(\mathrm{c}, \mathrm{d}, \frac{1}{\varrho} - 1) \tag{3.9}$$

$$\varrho \sim \mathcal{TPB}(e, f, \nu) \tag{3.10}$$

Its equivalent hierarchical structure is:

$$\Lambda_{\mathrm{j,k}} \sim \mathcal{N}(0, \theta_{j,k}) \tag{3.11}$$

$$\theta_{j,k} \sim \mathcal{G}\mathrm{a}(\mathrm{a}, \delta_{j,k}) \tag{3.12}$$

$$\delta_{j,k} \sim \mathcal{G}\mathrm{a}(\mathrm{b}, \phi_k) \tag{3.13}$$

$$\phi_k \sim \mathcal{G}\mathrm{a}(\mathrm{c}, \tau_k) \tag{3.14}$$

$$\tau_k \sim \mathcal{G}\mathrm{a}(\mathrm{d}, \eta) \tag{3.15}$$

$$\eta \sim \mathcal{G}\mathrm{a}(\mathrm{e}, \gamma) \tag{3.16}$$

$$\gamma \sim \mathcal{G}\mathrm{a}(\mathrm{f}, \nu) \tag{3.17}$$

The equivalence is based on the fact (see Appendix A 3.3, Proposition 1) that

$$\varphi \sim \mathcal{TPB}(a, b, \nu) \equiv \frac{\theta}{\nu} \sim \beta e'(a, b) \equiv \theta \sim \mathcal{G}\mathrm{a}(a, \delta) \text{and } \delta \sim \mathcal{G}\mathrm{a}(b, \nu), \tag{3.18}$$

where $\beta e'(a, b)$ and $\mathcal{G}\mathrm{a}$ indicate an inverse beta and a gamma distribution with shape and **rate** parameters. $\theta_{j,k}$ is generated from a mixture of sparse and dense components:

$$\theta_{j,k} \sim \pi \mathcal{G}\mathrm{a}\left(\mathrm{a}, \delta_{j,k}\right) + (1 - \pi)\delta(\phi_k), \tag{3.19}$$

where $\delta(\cdot)$ is the dirac delta function, and the hidden variable z_k, which indicates whether or not loading k is sparse (1) or dense (0), is generated from the following beta-Bernoulli distribution:

$$z_k | \pi \sim \mathcal{B}\mathrm{ern}(\pi), k = \{1,2, \dots, \mathcal{K}\} \tag{3.20}$$

$$\pi | \alpha, \beta \sim \beta e(\alpha, \beta) \tag{3.21}$$

Similarly, the hierarchical structure inducing sparsity in X , which is structurally identical to that for Λ, is written as:

$$\mathrm{X_{k,i}} \sim \mathcal{N}(0, \sigma_{k,i}) \tag{3.22}$$

$$\sigma_{k,i} \sim \pi\mathcal{G}\mathrm{a}\left(a_X, \rho_{k,i}\right) + (1-\pi)\delta(\omega_k) \tag{3.23}$$

$$\rho_{k,i} \sim \mathcal{G}\mathrm{a}(b_X, \omega_k) \tag{3.24}$$

$$\omega_k \sim \mathcal{G}\mathrm{a}(c_X, \kappa_k) \tag{3.25}$$

$$\kappa_k \sim \mathcal{G}\mathrm{a}(d_X, \chi) \tag{3.26}$$

$$\chi \sim \mathcal{G}\mathrm{a}(e_X, \varphi) \tag{3.27}$$

$$\varphi \sim \mathcal{G}\mathrm{a}(f_X, \xi) \tag{3.28}$$

with $\sigma_{k,i}$ generated from a two-component mixture. Here the hidden variable o_k, which indicates whether or not factor k is sparse (1) or dense (0), is generated from the following beta-Bernoulli distribution:

$$o_k|\, \pi_X \sim \mathcal{B}\mathrm{ern}(\pi_X), k = \{1,2, \dots, \mathcal{K}\} \tag{3.29}$$

$$\pi_X|\alpha_X, \beta_X \sim \beta e(\alpha_X, \beta_X). \tag{3.30}$$

Variational Expectation Maximation (VEM) approximation methods were used in *BicMix* to estimate values for latent variables and parameters directly from the data.

3.3. Variational EM

One approach to maximizing the likelihood function it to use iterative numerical optimization techniques. A general technique for finding maximum likelihood estimators in latent variable models is the expectation-maximization (EM) algorithm. A Model with latent variable Z can be represented by $p(X, Z, \theta)$, and its complete data set is $\{X, Z\}$. Since knowledge about latent variable Z is unknow and computing complete log likelihood $ln\, P\,(X, Z|\theta)$ is not straightforward, its expected value under the posterior distribution of the latent variable is considered, which correspond to E-step. In the M-step, the expectation of the complete-data log likelihood or log posterior if a prior $p(\theta)$is defined is computed. A general

EM algorithm is shown below:

1. Initial values for parameter θ

2. E-step:

   Evaluate posterior distribution of latent variable $p(Z|X,\theta^{old})$

3. M-step:

$$\theta^{new} = argmax_{\theta}\, Q(\theta,\theta^{old})$$

   where

$$Q(\theta,\theta^{old}) = E[\ln p(\theta|X,Z)] \propto E[\ln p(X,Z|\theta) + \ln p(\theta)]$$
$$= E[\ln p(X,Z|\theta)] + \ln p(\theta)$$
$$= \sum_{Z} p(Z|X,\theta^{old}) \times \ln p(X,Z|\theta) + \ln p(\theta)$$

   It's the expectation of complete-data log posterior.

4. If convergence threshold doesn't meet, let

$$\theta^{old} = \theta^{new}$$

   then go back to step 2.

It will become infeasible to evaluate the posterior distribution $p(Z|X,\theta)$ or to compute expectation with respect to this distribution when the dimensionality of the latent space is too high to work with directly or the posterior distribution has a highly complex form for which expectations are not analytically tractable. In such situations, we need to resort to approximation schemes. Laplace approximation and variational approximation are two widely used schemes. Variational approximation doesn't have normality assumption on the parameters as in Laplace approximation. *BicMix* used variational expectation maximization (VEM) for

parameter estimation. Detailed derivation of point estimates of the parameters in *BicMix* is listed in Appendix B.

For *BicMix* model, the posterior probability $\mathcal{P} = p(\Lambda, X, z, o, \Theta\,|Y)$ is written as:

$$\begin{aligned}\mathcal{P} \;&\propto p(Y|\Lambda, X)p(\Lambda\,|z, \Theta_\Lambda)p(z|\Theta_\Lambda)p(\Theta_\Lambda)p(X\,|o, \Theta_{\mathrm{X}})p(o|\Theta_{\mathrm{X}})p(\Theta_{\mathrm{X}}) \\ &= p(Y|\Lambda, X)\;\mathcal{P}(\Lambda)\mathcal{P}(\mathrm{X})\end{aligned} \tag{3.31}$$

where Θ_Λ and Θ_{X} are used to denote the set of parameters related to Λ and X, respectively. Then,

$$\begin{aligned}\mathcal{P}(\Lambda) =\; & p(\Lambda\,|z, \Theta_\Lambda)p(z|\Theta_\Lambda)p(\Theta_\Lambda) \\ &= \left[\prod_{j=1}^{P}\prod_{k=1}^{K}\mathcal{N}\left(\Lambda_{j,k}|\theta_{j,k}\right)\mathcal{G}a\left(\theta_{j,k}\,|a, \delta_{j,k}\right)\mathcal{G}a\left(\delta_{j,k}\,|b, \phi_k\right)\right]^{1_{z_k=1}} \\ &\quad \times \left[\prod_{j=1}^{P}\prod_{k=1}^{K}\mathcal{N}\left(\Lambda_{j,k}|\phi_k\right)\right]^{1_{z_k=0}} \\ &\times \left[\prod_{k=1}^{K}\mathcal{B}ern(z_k|\pi)\right]\mathcal{B}eta(\pi|\alpha, \beta)\end{aligned} \tag{3.32}$$

$$\times \left[\prod_{k=1}^{K}\mathcal{G}a(\phi_k\,|c, \tau_k)\mathcal{G}a(\,\tau_k\,|d, \eta)\right]\mathcal{G}a(\,\eta\,|e, \gamma)\mathcal{G}a(\,\gamma\,|f, \nu)$$

$$\begin{aligned}\mathcal{P}(\mathrm{X}) =\; & p(\mathrm{X}\,|o, \Theta_{\mathrm{X}})p(o|\Theta_{\mathrm{X}})p(\Theta_{\mathrm{X}}) \\ &= \left[\prod_{k=1}^{K}\prod_{i=1}^{n}\mathcal{N}\left(\mathrm{x}_{k,i}|\sigma_{k,i}\right)\mathcal{G}a\left(\sigma_{k,i}\,|a_{\mathrm{X}}, \rho_{k,i}\right)\mathcal{G}a\left(\rho_{k,i}\,|b_X, \omega_k\right)\right]^{1_{o_k=1}} \\ &\quad \times \left[\prod_{k=1}^{K}\prod_{i=1}^{n}\mathcal{N}\left(\mathrm{x}_{k,i}|\omega_k\right)\right]^{1_{o_k=0}}\end{aligned}$$

$$\times \left[\prod_{k=1}^{K} \mathcal{B}ern(o_k|\pi_X)\right] \mathcal{B}eta(\pi_X|\alpha_X, \beta_X)$$

$$\times \left[\prod_{k=1}^{K} \mathcal{G}a(\omega_k\ |c_X, \kappa_k)\mathcal{G}a(\ \kappa_k\ |d_X, \chi)\right] \mathcal{G}a(\ \chi\ |e_X, \varphi)\mathcal{G}a(\ \varphi\ |f_X, \xi)$$

Expected Complete log-likelihood for parameters related to Λ:

$$\Lambda\text{: } \mathbb{Q}(\Theta_\Lambda) = < \ell_c(\Theta_\Lambda, \Lambda|\mathrm{z}, \mathrm{X}, \mathrm{Y}) > \tag{3.33}$$

$$\begin{aligned}
\mathbb{Q}(\Theta_\Lambda) \ \propto\ & \langle \ln(p(Y|\ \Lambda, X, \Theta_\Lambda, \mathrm{z})) + \ln(\mathrm{p}(\Lambda|\mathrm{z}, \Theta_\Lambda) \times p(\mathrm{z}|\pi)) + \ \ln\ (p(\Theta_\Lambda) \\
& + \mathrm{l\,n}(p(\pi|\alpha, \beta))\rangle \\
\propto & -\frac{\boldsymbol{n}}{2}\ln(|\Psi|) - \sum_{j=1}^{P}\sum_{i=1}^{n} \frac{(y_{j,i} - \ \sum_{k=1}^{K} \Lambda_{j,k}\langle x_{k,i}\rangle)^2}{2\psi_{j,j}} \\
& + \sum_{k=1}^{K} \{\ \langle z_k\rangle \ln(\pi) + (1 - \langle z_k\rangle) \ln(1 - \pi)\} \\
+ \sum_{j=1}^{P}\sum_{k=1}^{K} & \langle z_k\rangle \left\{-\frac{1}{2}\ln\left(\theta_{j,k}\right) - \frac{{\Lambda_{j,k}}^2}{2\theta_{j,k}} + a\ln\left(\delta_{j,k}\right) + (a-1)\ln\left(\theta_{j,k}\right) - \delta_{j,k}\theta_{j,k}\right. \\
& \left. + b\ln(\phi_k)\ + (b-1)\ln\left(\delta_{j,k}\right) - \ \phi_k\delta_{j,k}\right\} \\
& + \sum_{j=1}^{P}\sum_{k=1}^{K} (1 - \langle z_k\rangle)\left\{-\frac{1}{2}\ln(\phi_k) - \frac{{\Lambda_{j,k}}^2}{2\phi_k}\right\} \\
& + \sum_{k=1}^{K} \{c\ln(\tau_k) + (c-1)\ln(\phi_k) - \tau_k\phi_k + d\ln(\eta) \\
& + (d-1)\ln(\tau_k) - \eta\tau_k\} + e\ln(\gamma) + (e-1)\ln(\eta) - \gamma\eta \\
& + f\ln(\nu) + (f-1)\ln(\gamma) - \nu\gamma\ +\ (\alpha-1)\ln(\pi) \\
& + (\beta-1)\ln(1-\pi)
\end{aligned} \tag{3.34}$$

Similarly, the expected complete log likelihood for parameters related to X takes the following form:

$$\begin{aligned}
\mathbb{Q}(\Theta_{\mathrm{X}}) \propto\ & \langle \ln(p(Y|\,\Lambda, X, \Theta_{\mathrm{X}}, \mathrm{o})) + \ln(\mathrm{p}(\mathrm{X}|\mathrm{o}, \Theta_{\mathrm{X}}) \times p(\mathrm{o}|\,\pi_{\mathrm{X}})) + \ln\,(p(\Theta_{\mathrm{X}}) \\
& + \ln(p(\pi_{\mathrm{X}}|\alpha_{\mathrm{X}}, \beta_{\mathrm{X}}))\rangle \\
\propto & -\frac{\boldsymbol{n}}{2}\ln(|\Psi|) - \sum_{j=1}^{P}\sum_{i=1}^{n}\frac{(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle)^2}{2\psi_{j,j}} \\
& + \sum_{k=1}^{K}\{\langle o_k\rangle \ln(\pi_{\mathrm{X}}) + (1 - \langle o_k\rangle)\ln(1 - \pi_{\mathrm{X}})\} \\
& + \sum_{k=1}^{K}\sum_{i=1}^{n}\langle o_k\rangle\left\{-\frac{1}{2}\ln(\sigma_{k,i}) - \frac{\langle x_{k,i}^2\rangle}{2\sigma_{k,i}} + a_{\mathrm{X}}\ln(\rho_{k,i}) + (a_{\mathrm{X}} - 1)\ln(\sigma_{k,i})\right. \\
& \left. - \rho_{k,i}\sigma_{k,i} + b_{\mathrm{X}}\ln(\omega_k) + (b_{\mathrm{X}} - 1)\ln(\rho_{k,i}) - \omega_k\rho_{k,i}\right\} \\
& + \sum_{k=1}^{K}\sum_{i=1}^{n}(1 - \langle o_k\rangle)\left\{-\frac{1}{2}\ln(\omega_k) - \frac{\langle x_{k,i}^2\rangle}{2\omega_k}\right\} \\
& + \sum_{k=1}^{K}\{c_{\mathrm{X}}\ln(\kappa_k) + (c_{\mathrm{X}} - 1)\ln(\omega_k) - \kappa_k\omega_k + d_{\mathrm{X}}\ln(\chi) \\
& + (d_{\mathrm{X}} - 1)\ln(\kappa_k) - \chi\kappa_k\} + e_{\mathrm{X}}\ln(\varphi) + (e_{\mathrm{X}} - 1)\ln(\chi) - \varphi\chi \\
& + f_{\mathrm{X}}\ln(\xi) + (f_{\mathrm{X}} - 1)\ln(\varphi) - \xi\varphi + (\alpha_{\mathrm{X}} - 1)\ln(\pi_{\mathrm{X}}) \\
& + (\beta_{\mathrm{X}} - 1)\ln(1 - \pi_{\mathrm{X}})
\end{aligned} \tag{3.35}$$

Computing the Maximum of A Posterior (MAP) estimates for the parameters that encourage sparsity in the $\boldsymbol{\Lambda}$ matrix,

$$\widehat{\Theta_{\Lambda}} = argmax_{\Theta_{\Lambda}}\,\mathbb{Q}(\Theta_{\Lambda}) \tag{3.36}$$

$$\frac{\partial\mathbb{Q}(\Theta_{\Lambda})}{\partial\Theta_{\Lambda}} = 0. \tag{3.37}$$

Derived estimates for the parameters are shown below, and the detailed derivation is proved in Appendix B.

$$\widehat{\boldsymbol{\Lambda}_{\boldsymbol{j},\cdot}} = \mathrm{y}_{\mathrm{j},\cdot}\psi_{jj}^{-1}\langle X\rangle^{T}\left(\langle X\psi_{jj}^{-1}X^{T}\rangle + \langle Z\rangle\Theta_{j,\cdot}{}^{-1} + (1-\langle Z\rangle)\Phi^{-1}\right)^{-1} \tag{3.38}$$

where

$$\Theta_{j,\cdot} = \begin{pmatrix} \theta_{j,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \theta_{j,K} \end{pmatrix} \quad \Phi = \begin{pmatrix} \phi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \phi_K \end{pmatrix} \quad \langle Z\rangle = \begin{pmatrix} \langle \mathrm{z}_1\rangle & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \langle \mathrm{z}_\mathrm{K}\rangle \end{pmatrix}. \tag{3.39}$$

$$\langle \boldsymbol{X}_{\cdot,\boldsymbol{i}}\rangle = (\Lambda^{T}\Psi^{-1}\Lambda + \langle \mathrm{O}\rangle\Sigma_i^{-1} + (\mathrm{I}-\langle \mathrm{O}\rangle)\Omega^{-1})^{-1}\Lambda^{T}\Psi^{-1}Y_{\cdot,i} \tag{3.40}$$

where

$$\Sigma_i = \begin{pmatrix} \sigma_{1,i} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{K,i} \end{pmatrix} \quad \Omega = \begin{pmatrix} \omega_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \omega_K \end{pmatrix} \quad \langle \mathrm{O}\rangle = \begin{pmatrix} \langle \mathrm{o}_1\rangle & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \langle \mathrm{o}_\mathrm{K}\rangle \end{pmatrix} \tag{3.41}$$

$$\begin{aligned} \widehat{\boldsymbol{\theta}_{\mathbf{j},\mathbf{k}}} &= \frac{(a-\frac{3}{2}) \pm \sqrt{(a-\frac{3}{2})^2 - 4\delta_{j,k}(-\frac{1}{2}{\Lambda_{j,k}}^2)}}{2\delta_{j,k}} \\ &= \frac{(2a-3) + \sqrt{(2a-3)^2 + 8{\Lambda_{j,k}}^2\delta_{j,k}}}{4\delta_{j,k}} \end{aligned} \tag{3.42}$$

$$\widehat{\delta_{\mathrm{j,k}}} = \frac{a+b-1}{\theta_{j,k}+\phi_k} \tag{3.43}$$

$$\widehat{\tau_k} = \frac{c+d-1}{\phi_k+\eta} \tag{3.44}$$

$$\hat{\eta} = \frac{Kd+e-1}{\gamma+\sum_{k=1}^{K}\tau_k} \tag{3.45}$$

$$\hat{\gamma} = \frac{e+f-1}{\eta+\nu} \tag{3.46}$$

$$\widehat{\phi_k} = \tfrac{H+\sqrt{H^2+MT}}{M}, \tag{3.47}$$

where

$$M = 2\left(\langle z_k\rangle \sum_{j=1}^{P} \delta_{j,k} + \tau_k\right), \tag{3.48}$$

$$H = \left(\langle z_k\rangle bP - \frac{(1-\langle z_k\rangle)P}{2} + (c-1)\right), \tag{3.49}$$

$$T = \left((1-\langle z_k\rangle)\sum_{j=1}^{P} {\Lambda_{j,k}}^2\right). \tag{3.50}$$

$$\langle z_k|\Theta_\Lambda\rangle$$

$$= \frac{\pi \prod_{j=1}^{P} \mathcal{N}\left(\Lambda_{j,k}|\theta_{j,k}\right)\mathcal{G}a\left(\theta_{j,k}\,|a,\delta_{j,k}\right)\mathcal{G}a\left(\delta_{j,k}\,|b,\phi_k\right)}{(1-\pi)\mathcal{N}\left(\Lambda_{j,k}|\phi_k\right) + \pi \prod_{j=1}^{P} \mathcal{N}\left(\Lambda_{j,k}|\theta_{j,k}\right)\mathcal{G}a\left(\theta_{j,k}\,|a,\delta_{j,k}\right)\mathcal{G}a\left(\delta_{j,k}\,|b,\phi_k\right)} \tag{3.51}$$

$$\widehat{\psi_{J,J}} = \frac{1+\frac{1}{2}\left(y_{j,i} - \sum_{k=1}^{K} \Lambda_{j,k}\langle x_{k,i}\rangle\right)^2}{n/2} = \frac{\left(y_{j,i} - \sum_{k=1}^{K} \Lambda_{j,k}\langle x_{k,i}\rangle\right)^2 + 2}{n}, \tag{3.52}$$

Its matrix form is:

$$\widehat{\Psi} = \frac{(Y - \Lambda\langle X\rangle)(Y - \Lambda\langle X\rangle)^T + 2\mathrm{I}}{n} = \frac{YY^T - 2Y\langle X^T\rangle\Lambda^T + \Lambda\langle X\rangle\langle X^T\rangle\Lambda^T + 2\mathrm{I}}{n} \tag{3.53}$$

4. NUMERICAL EXPERIMENTS

To reproduce the experimental results in original paper, we ran *BicMix* on both simulated data and the same real data as used in the paper. For simulated data, recovery & relevance score (Prelic et al, 2006) was used to measure the false discovery rate (FDR) and sensitivity in recovering true biclusterings, i.e., the average recovery quantifies how well each of the true biclusters is recovered by the biclustering algorithm, and the average bicluster relevance reflects to what extent the generated biclusters represent true biclusters. For real data, a breast cancer data set (Van et al, 2002), we computed distributions of number of genes and number of samples in estimated loadings and factors, respectively.

Experimental results show that our results are close to the original paper. We got very close R&R scores on simulated data shown in the first and second simulations of section 4.2, and got very similar distributions of number of genes and number of samples on breast cancer data shown in section 4.3. In addition to reproducing the experimental results in original paper, we ran extra simulations, the third and fourth ones in section 4.2, for assessing performance of *BicMix* under high noise and low noise with different number of true latent factors in a FA model. The simulation results exhibit good performance of *BicMix* model in discovering biclusters under different combinations of conditions.

In this chapter, we first introduced R&R score, followed by simulation on generated data and real data, respectively.

4.1. Recovery & Relevance Scores

Prelic et al. (2006) introduced general match scores -recovery and relevance scores- in order to assess the performances of the selected biclustering approaches. The recovery score is given by the function

$$REC = S_G(M_1, M_2) = \frac{\sum_{(G_1,C_1)\in M_1} \max_{(G_2,C_2)\in M_2} \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|}}{|M_1|} \quad (4.1)$$

and the relevance score is given by the function

$$REL = S_G(M_2, M_1) = \frac{\sum_{(G_2,C_2)\in M_2} \max_{(G_1,C_1)\in M_1} \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|}}{|M_2|} \quad (4.2)$$

where M_1 represents the true set of sparse biclusters (or matrices), M_2 represents the estimated set of sparse biclusters, (G_1, C_1) represents a true sparse bicluster composed of a set of genes and a set of conditions, and (G_2, C_2) represents an estimated sparse bicluster composed of a set of genes and a set of conditions. The functions show that conditions in a bicluster weren't considered while computing recovery and relevance score since these are gene match scores.

The following three specific cases represent that only one of the two scores are insufficient for evaluating the performance of a proposed biclustering approach.

Case 1: there are 15 true biclusters and 15 estimated biclusters that are identical to the 15 true biclusters. In this case, recovery score and relevance score are both 1.

Case 2: there are 15 true biclusters and 20 estimated biclusters that include the 15 true biclusters. In this case, recovery score is 1 but relevance score is $\frac{15}{20}$.

Case 3: there are 15 true biclusters and 10 estimated biclusters that are identical to 10 of the 15 true biclusters. In this case, recovery score is $\frac{10}{15}$ and relevance score is 1.

Therefore, recovery score and relevance score must team up in accessing the performance of a biclustering method.

4.2. Simulation

In this section, simulated data sets were generated and used to test validity of the biclustering model, *BicMix*. Simulated data were created for observation matrix

$$Y = \Lambda X + \varepsilon, \tag{4.3}$$

where Y has dimension $p = 500$ by $n = 300$ and $\varepsilon_{i,j} \sim \mathcal{N}(0, v^{-1})$. To simulate sparsity, for each loading and factor, a number $m \in [5, 20]$ of elements were randomly selected and assigned values drawn from $\mathcal{N}(0, \text{sd} = 2)$; the remaining elements were set to zero. For dense components, loadings and factors were drawn from a $\mathcal{N}(0, \text{sd} = 2)$ distribution. Components are allowed to share as many as five elements. Since non-zero values in loadings and factors were drawn from $\mathcal{N}(0, 2)$, noise $\varepsilon_{i,j} \sim \mathcal{N}(0, v^{-1})$ is considered as high noise when it follows $\mathcal{N}(0, \text{sd} = 2)$ and is considered as low noise when it follows $\mathcal{N}(0, \text{sd} = 1)$. We simulated six data sets with number of true latent factors as 15, 25, and 35 under low noise and high noise, respectively. Each simulated data set contains 10 sparse components in loading matrix and sparse matrix, respectively. *BicMix* was run 200 times on each simulated data set, and the results were used for testing validity of *BicMix*.

For the first simulation with low noise and 15 true latent factors, *BicMix* recovered the sparse loadings, sparse factors, and the biclusters well. Figure 4.1 shows distribution of computed 200 R&R scores, each corresponding to one run of BicMix, and it has consistent results as Figure 2(a) in original paper. Table 4.1 shows the summary of the R&R scores. R code for computing R&R scores can be found in Appendix C.

Table 4.1. Summary of R&R Scores with Low Noise and 15 True Latent Factors

	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Recovery Scores	0.5529	0.8748	0.9758	0.9337	0.9859	0.9859
Relevance Scores	0.8692	0.9751	0.9758	0.9760	0.9859	0.9893

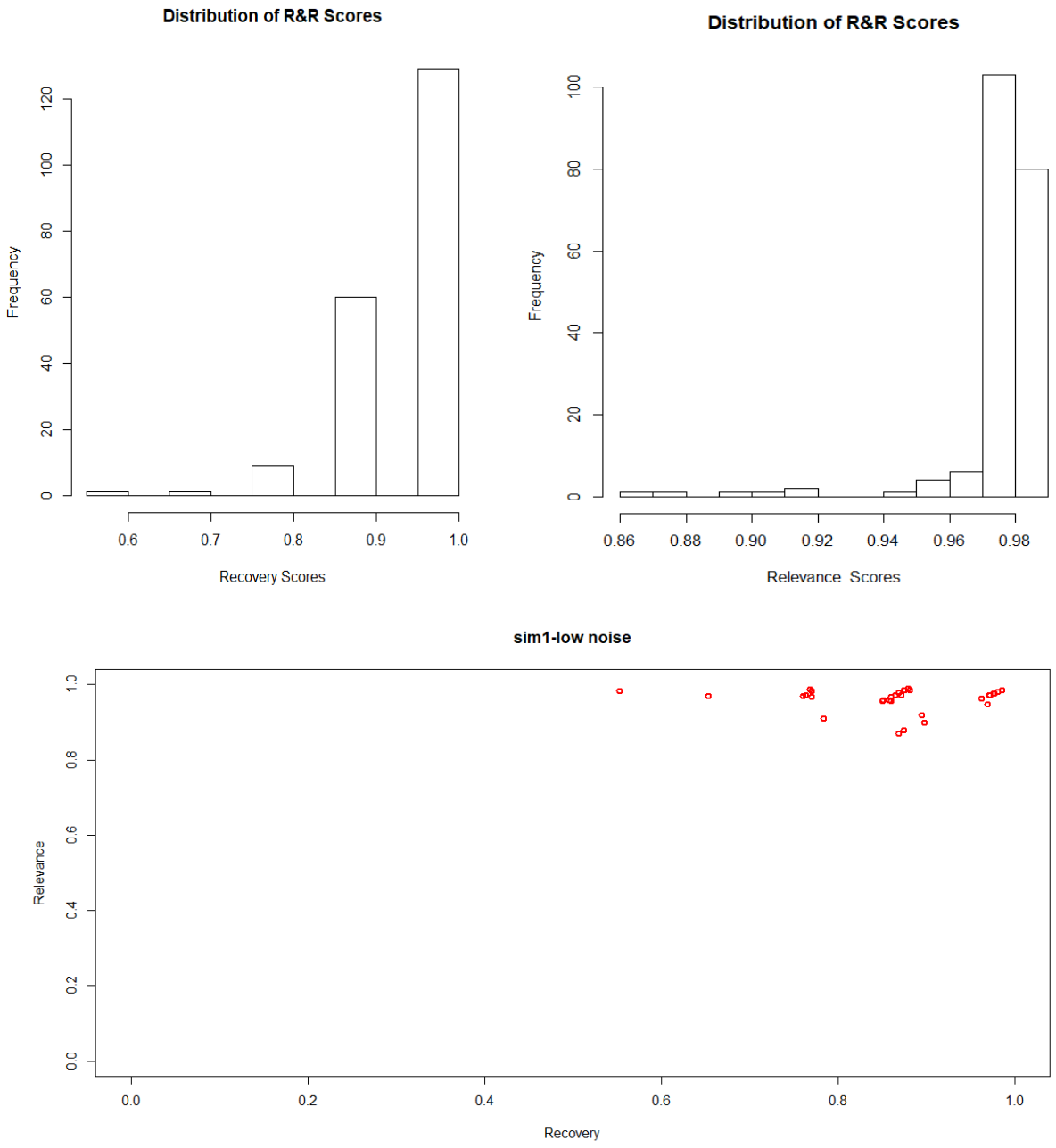

Figure 4.1. Distribution of Computed R&R Scores after Running BicMix 200 Times

For the second simulation, we ran *BicMix* on the simulated data set with high noise and 15 true latent factors, and compared its results with previous simulation. As shown in table 4.2, high noise increased false discovery rate.

For the third simulation, we ran *BicMix* on three simulated data sets with low noise and number of true latent factors as 15, 25, and 35, respectively. Table 4.3 shows larger number of

true latent factors got lower scores still at acceptable rate, more than 60% true clusters have been identified.

For the last simulation, we ran *BicMix* on three simulated data sets with high noise and number of true latent factors as 15, 25, and 35, respectively. Table 4.4 shows around 50% of true clusters have been identified in each case.

Table 4.2. R&R Scores Comparation between Low Noise and High Noise

	Low Noise	High Noise
Number of True Latent Factors	15	15
Sparse Components in Loading Matrix and Factor Matrix, Respectively	10	10
Number of True Clusters	9	9
Average Recovery Score	0.9337	0.6281
Average Relevance Score	0.9760	0.9809

Table 4.3. R&R Scores of Low Noise with Different Numbers of True Factors

	Low Noise		
Number of True Latent Factors	15	25	35
Sparse Components in Loading Matrix and Factor Matrix, Respectively	10	10	10
Number of True Clusters	9	10	10
Average Recovery Score	0.9337	0.6694	0.5985
Average Relevance Score	0.9760	0.9646	0.8259

Table 4.4. R&R Scores of High Noise with Different Numbers of True Factors

	High Noise		
Number of True Latent Factors	15	25	35
Sparse Components in Loading Matrix and Factor Matrix, Respectively	10	10	10
Number of True Clusters	6	3	2
Average Recovery Score	0.4527	0.34127	0.5336
Average Relevance Score	0.8483	0.73587	0.9974

4.3. Real Data

A breast cancer data set (Van et al, 2002) was used to estimate performance of *BicMix*. This data set contains 24,158 genes assayed in 337 breast tumor samples after removing gene that are > 10% missing and imputing missing values of included genes (Hastie et al, 1999). All patients in this data set had stage I or II breast cancer and were younger than 62 years old. Among the 337 patients, 193 had lymph-node negative disease and 144 had lymph-node positive disease; prognostic signatures such as BRCA1 mutations, estrogen receptor status (ER), distant metastasis free survival (DMFS) were collected for all patients.

We ran *BicMix* on these data, setting a = b = 0.5, c = 1, d = 0.5, e = f = 0.5 and $\nu = \xi = 1$ as in the simulations; the initial number of components was set to K = 300. Starting from 100 random values, *BicMix* was run until the total number of genes with non-zero loadings across components changed $\leq \frac{K_p}{100}$ over 100 iterations.

Distribution of the number of genes in estimated lambda matrix and distribution of number of samples in estimated factor matrix are shown in figure 4.2 and figure 4.3.

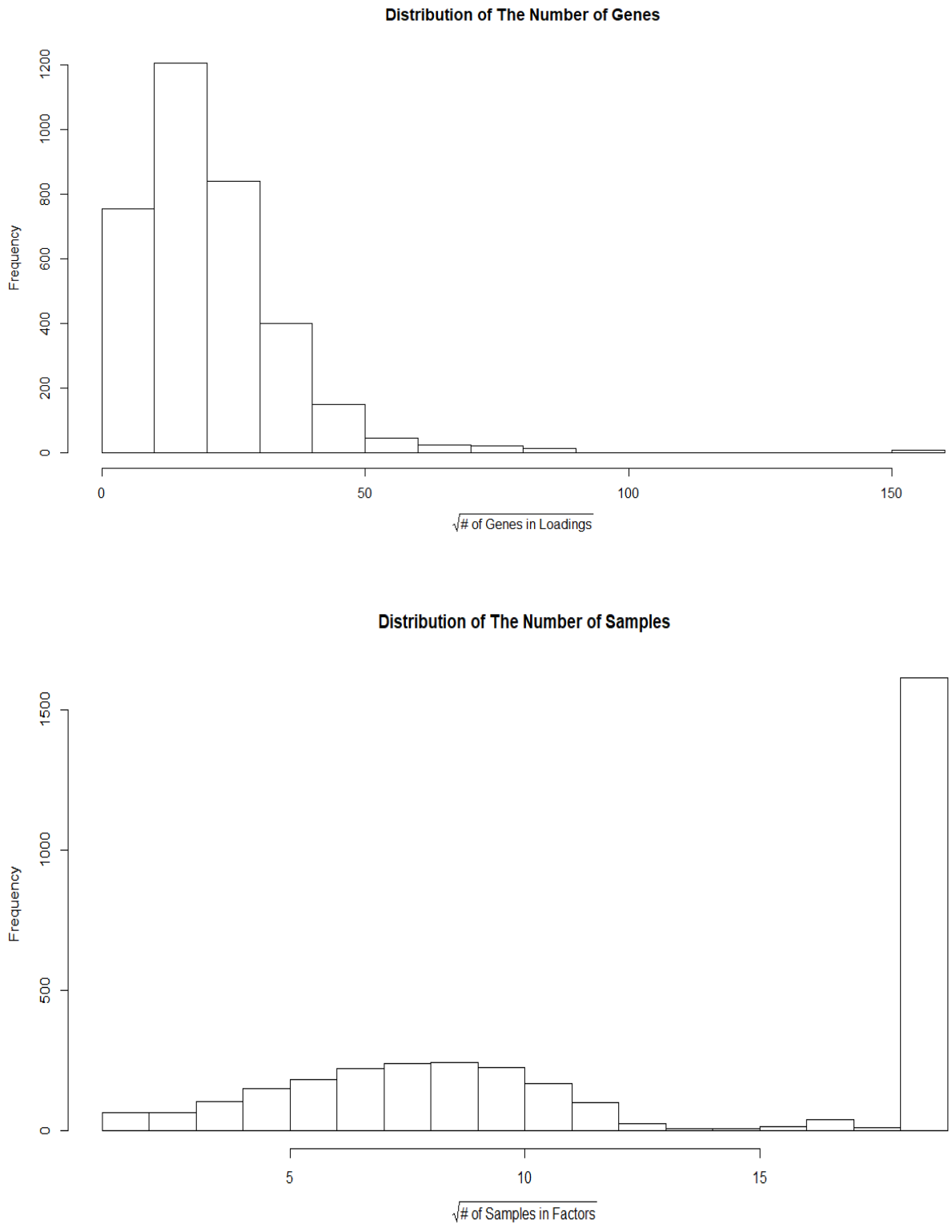

Figure 4.2. Histogram of the number of genes and samples in the breast cancer data

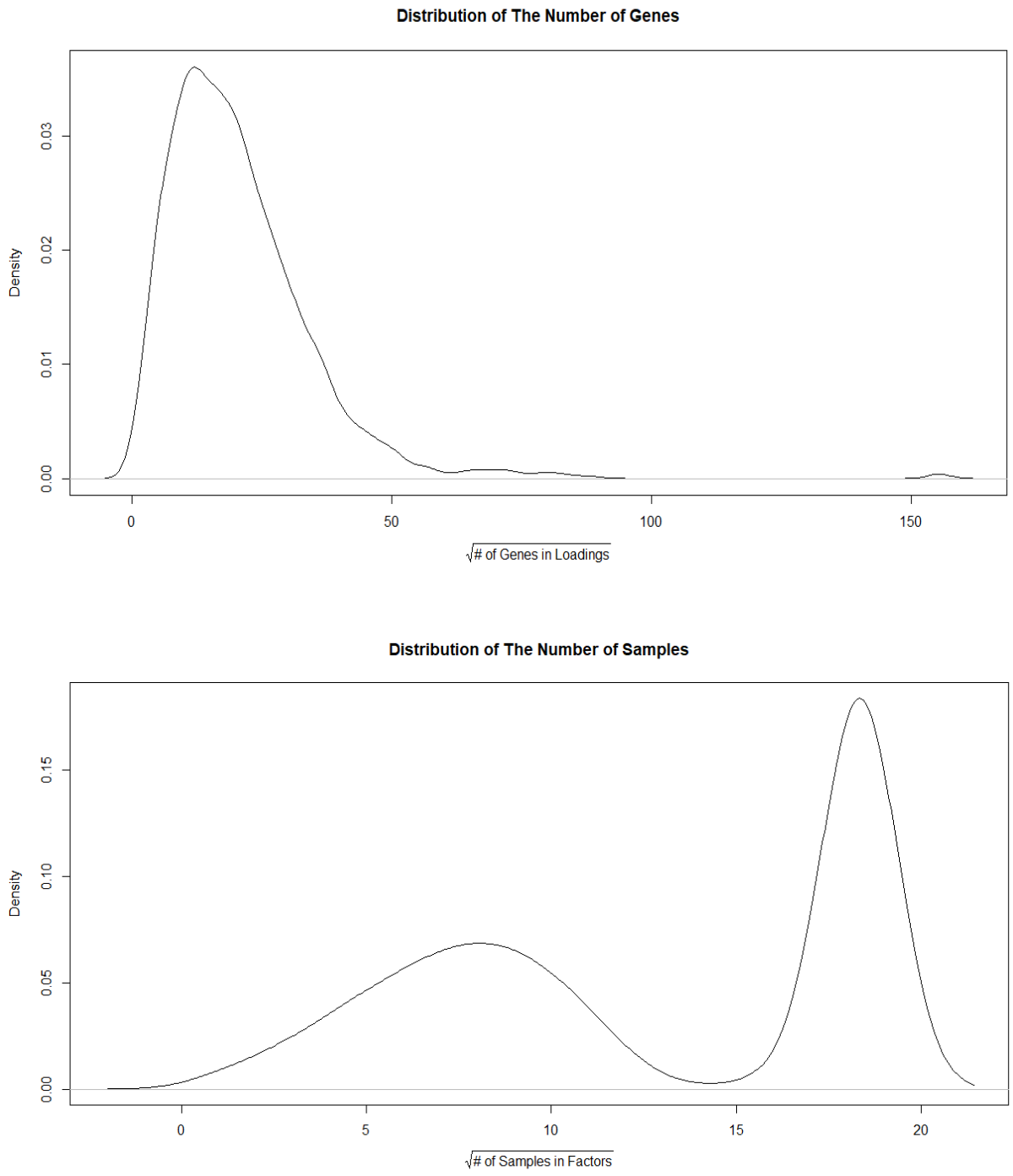

Figure 4.3. Density distribution of the number of genes and samples in the breast cancer data

Figure 4.3 shows that the number of genes in each sparse component was skewed to small numbers, and it shows consistent results as Figure 3(a, b) in the original paper.

5. DISCUSSION

A major contribution of this study was the comprehensive summary of fundamental techniques related to Bayesian sparse factor analysis along with proof of important formula and definitions in these techniques. It will make the understanding of Bayesian sparse Factor Analysis easier.

We explained how recovery and relevance scores were computed and explained using three specific examples why recovery score and relevance score must be used together to evaluate the performance of a proposed biclustering approach.

Detailed derivation of point estimates of the parameters in BicMix is provided in Appendix B, which is valuable in perceiving Variable EM algorithm and maximum a posterior technique.

BicMix adopted the recovery and relevance score (Prelic et al, 2006) to measure the false discovery rate (FDR) and sensitivity in recovering true biclusters. However, this measurement system only considered genes while computing recovery score and relevance score (see 4.1). Since BicMix imposed sparse priors on factors besides on loadings, it makes more sense to include conditions while computing the match scores. This will be our future work.

REFERENCES

[1] Gao, C., McDowell, I.C., Zhao, S. and Brown, C.D. (2016). Context specific and differential gene coexpression networks via Bayesian biclustering. PLoS Computational Biology 2016, 12(7):e1004791.

[2] Griffin, J. E. and Brown, P. J. (2010). Inference with normal-gamma prior distributions in regression problems. Bayesian Analysis, 5: 171–188.

[3] West, M. (2003). Bayesian factor regression models in the "large p, small n" paradigm. Bayesian Statistics 7: 723–732.

[4] Armagan, A., Dunson, D.B. and Clyde, M. (2011). Generalized beta mixtures of Gaussians. In: Proceedings of Neural Information Processing Systems. pp. 523–531.

[5] Carvalho, C.M., Polson, N.G. and Scott, J.G. (2010). The horseshoe estimator for sparse signals. Biometrika 97: 465–480. doi: 10.1093/biomet/asq017.

[6] Van't Veer LJ, Dai, H., Van de Vijver MJ, He, Y.D., Hart, AAM, et al. (2002). Gene expression profiling predicts clinical outcome of breast cancer. Nature 415: 530–536. doi: 10.1038/415530a

[7] van de Vijver MJ, He, Y.D., van't Veer LJ, Dai, H., Hart, AAM, et al. (2002). A gene-expression signature as a predictor of survival in breast cancer. The New England Journal of Medicine 347: 1999–2009. doi: 10.1056/NEJMoa021967 PMID: 12490681

[8] Hastie, T., Tibshirani, R., Sherlock, G., Eisen, M., Brown, P., et al. (1999). Imputing missing data for gene expression arrays. Technical report.

[9] Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., et al. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. Bioinformatics 22: 1122–1129. doi: 10. 1093/bioinformatics/btl060 PMID: 16500941.

[10] Blum, Y., Mignon, G.L., Lagarrigue, S. and Causeur, D. (2010). A factor model to analyze heterogeneity in gene expression. BMC bioinformatics, 11(1):368.

[11] Lucas, J.E., Kung, H.N. and Chi, J.T. ((2010). Latent Factor Analysis to Discover Pathway Associated Putative Segmental Aneuploidies in Human Cancers. PLoS Comput Biol, 6(9):e1000920.

[12] Engelhardt, B.E. and Stephens, M. (2010). Analysis of population structure: A unifying framework and novel methods based on sparse factor analysis. PLoS Genetics, 6(9): e1001117.

[13] Carvalho, C.M., Lucas, J.E., Wang, Q., Chang, J., Nevins, J.R. and West, M. (2008). High-dimensional sparse factor modelling - applications in gene expression genomics. Journal of the American Statistical Association, 103:1438{1456, PMCID 3017385}.

[14] Gao, C., Brown, C.D. and Engelhardt, B.E. (2013). A latent factor model with a mixture of sparse and dense factors to model gene expression data with confounding effects. arXiv:1310.4792.

[15] Hochreiter, S., Bodenhofer, U., Heusel, M., Mayr, A., Mitterecker, A., et al. (2010). FABIA: Factor analysis for bicluster acquisition. Bioinformatics 26: 1520–1527. doi: 10.1093/bioinformatics/btq227 PMID:20418340.

[16] Ben-Dor, A., Chor, B., Karp, R. and Yakhini, Z. (2003). Discovering local structure in gene expression data: The order-preserving submatrix problem. Journal of Computational Biology 10: 373–384. doi: 10.1089/ 10665270360688075 PMID: 12935334

[17] Murali, T.M. and Kasif, S. (2003). Extracting conserved gene expression motifs from gene expression data. Proceedings of the Pacific Symposium on Biocomputing: 77–88.

[18] Li, G., Ma, Q., Tang, H., Paterson, A.H. and Xu, Y. (2009). QUBIC: A qualitative biclustering algorithm for analyses of gene expression data. Nucleic Acids Research 37: e101–e101. doi: 10.1093/nar/gkp491 PMID: 19509312

[19] Hyvärinen, A. (1999). Survey on independent component analysis. Neural Comput. Surv.,2, 94–128.

[20] Engelhardt, B. and Stephens, M. (2010). Analysis of population structure: a unifying framework and novel methods based on Sparse Factor Analysis PLoS Genet 6:–e1001117

[21] Kaiser, H.F. (1958). The varimax criterion for analytic rotation in factor analysis. Psychometrika, 23(3):187.

[22] Carvalho, C.M., Polson, N.G. and Scott, J.G. (2010). The horseshoe estimator for sparse signals. Biometrika 97: 465–480. doi: 10.1093/biomet/asq017.

[23] Sugiyama, M. (2015). Introduction to Statistical Machine Learning. Morgan Kaufmann, ISBN-10: 9780128021217

APPENDIX A. FUNDAMENTAL TECHNIQUES OF SPARSITY INDUCING METHODS

A.1. Least Squares (LS) Regression

Linear least squares regression is a widely used modeling method. It's a procedure to determine the best fit line to observed data. Generalized form of linear least squares method (Sugiyama, M., 2015) is

$$f_{\Theta}(x;\ \theta) = \sum_{i=1}^{b} \theta_i\, \phi_i(x) = \theta^T \phi(x). \tag{A.1}$$

It is a linear combination of b unknown parameters. Linear least squares regression gets its name from the way how unknow parameters are estimated. The training least squares (LS) error J_{LS} is expressed as:

$$J_{LS} = \frac{1}{2}||Y - \Phi\Theta||^2 \tag{A.2}$$

where

$$Y = (y_i, \cdots, y_n)^T, \Theta = (\theta_i, \cdots, \theta_b)^T, and\ \Phi = \begin{pmatrix} \phi_1(x_1) & \cdots & \phi_b(x_1) \\ \vdots & \ddots & \vdots \\ \phi_1(x_n) & \cdots & \phi_b(x_n) \end{pmatrix}_{nxb}. \tag{A.3}$$

Note, number $\frac{1}{2}$ in J_{LS} is merely for convenience and it won't affect the generation of optimal parameters.

Partial derivative of J_{LS} w.r.t parameter Θ:

$$\begin{aligned} J_{LS} = \frac{1}{2}\left|\left|Y - \Phi\Theta\right|\right|^2 &= \frac{1}{2}(Y - \Phi\Theta)^T(Y - \Phi\Theta) \\ &= \frac{1}{2}(Y^TY - Y^T\Phi\Theta - \Theta^T\Phi^TY + \Theta^T\Phi^T\Phi\Theta) \\ &= \frac{1}{2}Tr(Y^TY - Y^T\Phi\Theta - \Theta^T\Phi^TY + \Theta^T\Phi^T\Phi\Theta) \end{aligned} \tag{A.4}$$

$$\nabla_{\Theta} J_{LS} = \frac{1}{2}(-\Phi^TY - \Phi^TY + \Phi^T\Phi\Theta + \Phi^T\Phi\Theta) = \Phi^T\Phi\Theta - \Phi^TY = 0 \tag{A.5}$$

$$\widehat{\Theta_{LS}} = (\Phi^T\Phi)^{-1}\Phi^T Y \tag{A.6}$$

Note: the following two formulas has been used in deriving $\widehat{\Theta_{LS}}$:

$$\frac{\partial Tr(AXB)}{\partial X} = A^T B^T \ , \qquad \frac{\partial Tr(AX^T B)}{\partial X} = BA$$

A.2. Constrained LS Regressions

One of the limits of ordinary least squares regression is making the model overfitted on the training data and therefore has poor performance on prediction. Adding constraints to the calculation of training least squares error is a valid approach to avoid overfitting. In this section, two different constrained least squares methods are briefly introduced: $\mathcal{L}_2$ constrained least squares and $\mathcal{L}_1$ constrained least squares.

A.2.1. $\mathcal{L}_2$ Constraint LS Regression

$$\begin{gathered} min_\Theta J_{LS}(\Theta) \\ subject\ to\ ||\Theta||^2 \leq R^2 \end{gathered} \tag{A.7}$$

Its Lagrange dual problem is given as:

$$\begin{gathered} max_\lambda\, min_\Theta \ \left[J_{LS}(\Theta) + \frac{\lambda}{2}\left(||\Theta||^2 - R^2\right)\right] \\ subject\ to\ \lambda \geq 0 \end{gathered} \tag{A.8}$$

The solution of $\mathcal{L}_2$ constrained least squares is

$$\text{argmin}_\Theta \left[J_{LS}(\Theta) + \frac{\lambda}{2}(||\Theta||^2 - R^2)\right] = \text{argmin}_\Theta \left[J_{LS}(\Theta) + \frac{\lambda}{2}||\Theta||^2\right]$$

$$\begin{gathered} \frac{\partial\left[J_{LS}(\Theta) + \frac{\lambda}{2}||\Theta||^2\right]}{\partial\Theta} = \Phi^T\Phi\Theta - \Phi^T Y + \frac{\lambda}{2}2\Theta = 0 \\ \widehat{\Theta_{CLS}} = (\Phi^T\Phi + \lambda\Theta)^{-1}\Phi^T Y \end{gathered} \tag{A.9}$$

where

$$\begin{cases} ||\Theta||^2: & regularizer \\ \lambda: & regularization\ parameter \end{cases}$$

Note: for a Lagrange dual problem

$$min_x f(x)\ \ subject\ \ g(x) \leq 0,$$

its Lagrange dual problem is given as:

$$max_\lambda min_x L(x,\lambda)\ \ subject\ \ \ \lambda \geq 0$$

where

$$L(x,\lambda) = f(x) + \lambda^T g(x),\ \ \lambda = \left(\lambda_1, \cdots, \lambda_p\right)^T$$

A.2.2. $\mathcal{L}_1$ Constraint LS Regression

This method is also called ordinary LASSO (Least Absolute Selection and Shrinkage Operator). It uses an $\mathcal{L}_1$ regularization penalty to achieve sparsity in regression.

$$min_\Theta\, J_{LS}(\Theta), \qquad subject\ to\ ||\Theta||_1\ \leq\ R^2 \tag{A.10}$$

where $||\Theta||_1 = \sum_{j=1}^{b} |\theta_j|$.

Its Lagrange dual problem is given as:

$$max_\lambda\, min_\Theta\ \ \left[J_{LS}(\Theta) + \lambda\left(\left|||\Theta|\right|_1 -\ R^2\right)\right],\ \ \ subject\ to\ \lambda\ \geq 0. \tag{A.11}$$

Estimated parameters can be computed by solving $argmin_\Theta\ [J_{LS}(\Theta) + \lambda||\Theta||_1]$.

Since $\mathcal{L}_1 - CLS$ is not differential at the origin, solving it is not as straight forward as $\mathcal{L}_2 - CLS$. There are different ways to solve this problem. Alternating direction method of multipliers (ADMM) is one of those approaches.

The figure below graphically shows $\mathcal{L}_1$ and $\mathcal{L}_2$ constrained LS, respectively.

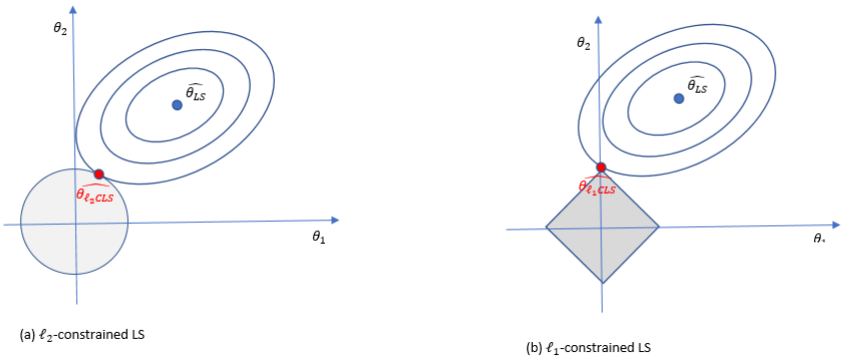

Figure A.1. $\mathcal{L}_1$ and $\mathcal{L}_2$ constrained LS

A.2.3. Bayesian Lasso Regression

Lasso estimates can be interpreted as posterior mode estimates when the regression parameters have independent and identical Laplace (i.e., double-exponential) priors. Specifically, the lasso estimate can be viewed as the mode of the posterior distribution of β

$$\widehat{\beta_L} = argmax_\beta \, P(\beta|y, \sigma^2, \tau) \tag{A.11}$$

where

$$(\beta|\tau\,) \sim Laplace(\mu = 0, \tau), \tag{A.12}$$

$$p(\beta|\tau\,) = \left(\frac{\tau}{2}\right)^p \exp(-\tau||\beta||_1), \tag{A.13}$$

and the likelihood

$$(y|\beta, \sigma^2) \sim \mathcal{N}(X\beta, \sigma^2 I_n). \tag{A.14}$$

For any fixed values σ^2 and $\tau > 0$, the posterior mode of β is the ordinary lasso estimate with penalty $\lambda = 2\tau\sigma^2$, and its proof is shown below.

Proof:

$$P(\beta|y, \sigma^2, \tau) \propto P(\beta|\tau)P(y|\beta, \sigma^2)$$

$$\propto \exp\left(-\tau\left|\left|\beta\right|\right|_1\right) \exp\left(-\frac{(y - X\beta)^2}{2\,\sigma^2}\right)$$

$$\propto \exp\left(-\left(\frac{(y-X\beta)^2}{2\,\sigma^2}+\tau\left|\left|\beta\right|\right|_1\right)\right)$$

$$\widehat{\beta_L} = argmax_\beta\, P(\beta|y,\sigma^2,\tau)$$

$$= argmax_\beta \exp\left(-\left(\frac{(y-X\beta)^2}{2\,\sigma^2}+\tau\left|\left|\beta\right|\right|_1\right)\right)$$

$$= argmin_\beta\left(\frac{(y-X\beta)^2}{2\,\sigma^2}+\tau\left|\left|\beta\right|\right|_1\right)$$

$$= argmin_\beta\left((y-X\beta)^2+2\tau\sigma^2\left|\left|\beta\right|\right|_1\right)$$

recall, ordinary lasso

$$\text{argmin}_\beta(J_{LS}(\beta)+\lambda||\beta||_1) = \text{argmin}_\beta((y-X\beta)^2+\lambda||\beta||_1).$$

Therefore, as $\lambda = 2\tau\sigma^2$

$$argmin_\beta\left((y-X\beta)^2+2\tau\sigma^2\left|\left|\beta\right|\right|_1\right) = \text{argmin}_\beta((y-X\beta)^2+\lambda||\beta||_1) \quad \blacksquare$$

A.3. Sparse-Inducing Priors

A.3.1. Laplace Prior

A Laplace distribution can be represented by a scale mixture of Gaussians, i.e.,

$$\beta \sim Laplace\left(a=0, b=\sqrt{\frac{D}{\alpha}}\right) \equiv \beta \sim N(0, 2D\tau),\ \tau \sim Exp(\alpha). \qquad \text{(A.15)}$$

To proof above identical representation of Laplace distribution, we computed moment generating functions (MGFs) of these two different representations and their MGFs shows equal. Detail of the proof is shown below.

Proof:

Let $\boldsymbol{\beta \sim Laplace(a,b)}$, its PDF is

$$f(\beta) = \frac{1}{2b}exp\left(\frac{-|\beta-a|}{b}\right), x \in R$$

where a is a location parameter (u), and $b > 0$. The moment generating function (MGF) of β is

$$M_\beta(t) = E\left[e^{t\beta}\right] = \int_{-\infty}^{\infty} e^{t\beta} \frac{1}{2b} exp\left(\frac{-|\beta - a|}{b}\right) d\beta.$$

Let $y = \frac{\beta - a}{b}$, then $\beta = yb + a$, and $d\beta = bdy$.

$$M_\beta(t) = \frac{1}{2b} \int_{-\infty}^{\infty} e^{t(yb+a)}\, e^{-|y|} b dy$$

$$= \frac{1}{2} e^{at} \left[\int_{-\infty}^{0} e^{(bt+1)y}\, dy + \int_{0}^{\infty} e^{(bt-1)y}\, dy\right]$$

$$= \frac{1}{2} e^{at} \left[\frac{1}{bt+1}(1-0) + \frac{1}{bt-1}(0-1)\right] \quad , \qquad -\frac{1}{b} < t < \frac{1}{b}$$

$$= \frac{e^{at}}{1 - t^2 b^2}$$

Let $\boldsymbol{\beta} \sim \boldsymbol{N(0, 2D\tau)}$, $\boldsymbol{\tau} \sim \boldsymbol{Exp(\alpha)}$, their PDF are

$$f_\beta(\beta|\tau) = \frac{1}{\sqrt{4\pi D\tau}} e^{-\frac{\beta^2}{4D\tau}}, \text{ and}$$

$$f_\tau(\tau; \alpha) = \alpha e^{-\alpha\tau}$$

The marginal PDF of β is:

$$f_\beta(\beta) = \int_0^\infty f_\beta(\beta, \tau) d\tau = \int_0^\infty f_\beta(\beta|\tau) f_\tau(\tau; \alpha) d\tau = \int_0^\infty \frac{1}{\sqrt{4\pi D\tau}} e^{-\frac{\beta^2}{4D\tau}} \times \alpha e^{-\alpha\tau} d\tau$$

This integral involves Gauss error functions, so we'll turn to looking at MGF.

$$M_\beta(t) = E\left[e^{t\beta}\right] = \int_{-\infty}^{\infty} e^{t\beta} \times f_\beta(\beta) d\beta$$

$$= \int_{-\infty}^{\infty} e^{t\beta}\, d\beta \int_0^\infty \frac{1}{\sqrt{4\pi D\tau}} e^{-\frac{\beta^2}{4D\tau}} \times \alpha e^{-\alpha\tau} d\tau$$

$$= \int_0^\infty \alpha e^{-\alpha\tau} d\tau \int_{-\infty}^{\infty} \frac{1}{\sqrt{4\pi D\tau}} e^{-\frac{\beta^2}{4D\tau}} \times e^{t\beta} d\beta$$

$$= \int_0^\infty \alpha e^{-\alpha\tau} d\tau \int_{-\infty}^\infty \frac{1}{\sqrt{4\pi D\tau}} e^{-\left(\frac{(\beta-2D\tau t)^2}{4D\tau} - \frac{4D^2\tau^2 t^2}{4D\tau}\right)} d\beta$$

$$= \int_0^\infty \alpha e^{-\alpha\tau} e^{D\tau t^2} d\tau \int_{-\infty}^\infty \frac{1}{\sqrt{4\pi D\tau}} e^{-\frac{(\beta-2D\tau t)^2}{4D\tau}} d\beta$$

$$= \int_0^\infty \alpha e^{-\alpha\tau} e^{D\tau t^2} d\tau \times 1$$

$$= \int_0^\infty \alpha e^{(Dt^2-\alpha)\tau} d\tau$$

$$= \frac{\alpha}{Dt^2 - \alpha} (0 - 1), \quad t^2 \leq \frac{\alpha}{D}$$

$$= \frac{1}{1 - \frac{D}{\alpha} t^2}$$

Recall, the derived MGF of Laplace (a, b) is

$$M_\beta(t) = \frac{e^{at}}{1 - t^2 b^2}.$$

Let $a = 0$ and $b = \sqrt{\frac{D}{\alpha}}$, then

$$M_\beta(t) = \frac{e^{at}}{1 - t^2 b^2} = \frac{1}{1 - \frac{D}{\alpha} t^2}$$

It shows that $Laplace(a, b)$ and scale mixture Gaussians $\beta \sim N(0, 2D\tau), \tau \sim Exp(\alpha)$ have the same MGF when $a = 0 \; and \; b = \sqrt{\frac{D}{\alpha}}$. Their pdf is

$$f_\beta\left(\beta; a = 0, b = \sqrt{\frac{D}{\alpha}}\right) = \frac{1}{2b} \exp\left(-\frac{|\beta - a|}{b}\right) = \frac{1}{2}\sqrt{\frac{\alpha}{D}} \exp\left(-\sqrt{\frac{\alpha}{D}} |\beta|\right).$$

Therefore,

$$\beta \sim Laplace\left(a = 0, b = \sqrt{\frac{D}{\alpha}}\right) \equiv \beta \sim N(0, 2D\tau), \qquad \tau \sim Exp(\alpha) \qquad \blacksquare$$

A.3.2. Horseshoe Prior

In a simple situation where $(y|\beta) \sim N(\beta, \sigma^2 I)$ and where β is believed to be sparse. The horseshoe prior assumes that each β_i is conditionally independent with density $\pi_{HS}(\beta_i|\tau)$, where π_{HS}can be represented as a scale mixture of Gaussians:

$$\begin{aligned} (\beta_i|\lambda_i, \tau) &\sim N\left(0, {\lambda_i}^2\tau^2\right) \\ \lambda_i &\sim C^{+(0,1)} \end{aligned} \tag{A.16}$$

where $C^{+(0,1)}$ is a half-Cauchy distribution for the standard deviation λ_i. Here, λ_i is local shrinkage parameters and τ is a global shrinkage parameter. The density π_{HS} is perfectly well defined without reference to the λ_i's, which can be marginalized away. But by writing the horseshoe prior as a scale mixture of Gaussians, we can identify its relationship with commonly used procedures in supervised learning, e.g., exponential mixing, with ${\lambda_i}^2 \sim exp(\alpha)$, implies independent Laplacian priors for each β_i; inverse-gamma mixing, with ${\lambda_i}^2 \sim IG(a, b)$, leads to $student - t$ priors.

Assuming $\sigma^2 = 1$, then

$$\begin{aligned} (y_i\ |\beta_i)\ &\sim N(\beta_i, 1), \text{ and} \\ (\beta_i|\lambda_i\tau)\ &\sim N(0, {\lambda_i}^2\tau^2). \end{aligned} \tag{A.17}$$

Their pdfs are:

$$\begin{aligned} p(y_i\ |\beta_i)\ &\propto \exp\left(-\tfrac{(y_i-\beta_i)^2}{2}\right), \text{ and} \\ p(\beta_i\ |\lambda_i\tau)\ &\propto \tfrac{1}{\lambda_i\tau} exp\left(-\tfrac{{\beta_i}^2}{2{\lambda_i}^2\tau^2}\right). \end{aligned} \tag{A.18}$$

Posterior of $\left(\beta_i|y_i, {\lambda_i}^2\tau^2\right)$ can be derived as

$$p\left(\beta_i|y_i, {\lambda_i}^2\tau^2\right) \propto \frac{1}{\lambda_i} exp\left(-\frac{{\beta_i}^2}{2{\lambda_i}^2\tau^2}\right) \times \exp\left(-\frac{(y_i-\beta_i)^2}{2}\right) \tag{A.19}$$

$$\propto \frac{1}{\lambda_i} exp\left(-\frac{1}{2}\left((y_i-\beta_i)^2+\frac{\beta_i^2}{\lambda_i^2\tau^2}\right)\right).$$

Now let's consider exponent part only, i.e.,

$$(y_i-\beta_i)^2+\frac{\beta_i^2}{\lambda_i^2\tau^2} = y_i^2+\beta_i^2-2y_i\beta_i+\frac{\beta_i^2}{\lambda_i^2\tau^2}$$

$$= \beta_i^2\left(1+\frac{1}{\lambda_i^2\tau^2}\right)-2y_i\beta_i+y_i^2$$

$$= \left(1+\frac{1}{\lambda_i^2\tau^2}\right)\left[\beta_i^2-\frac{2y_i}{1+\frac{1}{\lambda_i^2\tau^2}}\beta_i+\frac{y_i^2}{1+\frac{1}{\lambda_i^2\tau^2}}\right]$$

$$= \left(1+\frac{1}{\lambda_i^2\tau^2}\right)\left[\left(\beta_i-\frac{y_i}{1+\frac{1}{\lambda_i^2\tau^2}}\right)^2-\left(\frac{y_i}{1+\frac{1}{\lambda_i^2\tau^2}}\right)^2+\frac{y_i^2}{1+\frac{1}{\lambda_i^2\tau^2}}\right] \quad \text{(A.20)}$$

$$= \left(1+\frac{1}{\lambda_i^2\tau^2}\right)\left[\left(\beta_i-\frac{y_i}{1+\frac{1}{\lambda_i^2\tau^2}}\right)^2\right]+const.$$

Therefore,

$$p\left(\beta_i|y_i,{\lambda_i}^2\tau^2\right) \propto exp\left(-\frac{\left(\beta_i-\frac{y_i}{1+\frac{1}{\lambda_i^2\tau^2}}\right)^2}{2\left(\frac{1}{1+\frac{1}{\lambda_i^2\tau^2}}\right)}\right)$$

$$\sim N\left(\frac{y_i}{1+\frac{1}{\lambda_i^2\tau^2}},\quad \frac{1}{1+\frac{1}{\lambda_i^2\tau^2}}\right) \tag{A.21}$$

$$= N\left(\frac{\lambda_i^2\tau^2}{1+\lambda_i^2\tau^2}y_i,\quad \frac{\lambda_i^2\tau^2}{1+\lambda_i^2\tau^2}\right)$$

So, mean of $\left(\beta_i|y_i,{\lambda_i}^2\tau^2\right)$:

$$E\left[\beta_i\middle|y_i,{\lambda_i}^2\tau^2\right] = \frac{\lambda_i^2\tau^2}{1+\lambda_i^2\tau^2}y_i$$

$$= \left(1-\frac{1}{1+\lambda_i^2\tau^2}\right)y_i + \left(\frac{1}{1+\lambda_i^2\tau^2}\right)\times 0 \tag{A.22}$$

$$= (1-k_i)y_i$$

where

$$k_i = \frac{1}{1+\lambda_i^2\tau^2}. \tag{A.23}$$

Variable k_i is a random shrinkage coefficient, the amount of weight that posterior mean for β_i places on 0 once y has been observed. The prior on k_i is given by:

$$P_k(k_i\,;\,\tau) = \frac{\tau}{\pi}\frac{1}{1-(1-\tau^2)k_i}(1-k_i)^{-\frac{1}{2}}{k_i}^{-\frac{1}{2}}, \tag{A.24}$$

and its proof is shown below.

Proof: $P_k(k_i\,;\,\tau) = \frac{\tau}{\pi}\frac{1}{1-(1-\tau^2)k_i}(1-k_i)^{-\frac{1}{2}}{k_i}^{-\frac{1}{2}}$

$$\lambda_i \sim C^+(0,1),\quad half-Cauchy$$

$$P_\lambda(\lambda_i) = \frac{2}{\pi}\left(\frac{1}{{\lambda_i}^2 + 1}\right)$$

Let $k_i = \frac{1}{1+\lambda_i^2\tau^2}$

$$\lambda_i = \frac{1}{\tau}\left(\frac{1}{k_i} - 1\right)^{\frac{1}{2}}$$

Jacobian:

$$\frac{\partial \lambda_i}{\partial k_i} = -\frac{1}{2\tau k_i}(1-k_i)^{-\frac{1}{2}}{k_i}^{-\frac{1}{2}}$$

transformation,

$$P_k(k_i\,;\tau) = P_\lambda\left(\frac{1}{\tau}\left(\frac{1}{k_i} - 1\right)^{\frac{1}{2}}\right)\left|\frac{\partial \lambda_i}{\partial k_i}\right|$$

$$= \frac{2}{\pi}\left(\frac{1}{\left(\frac{1}{\tau}\left(\frac{1}{k_i} - 1\right)^{\frac{1}{2}}\right)^2 + 1}\right) \times \frac{1}{2\tau k_i}(1-k_i)^{-\frac{1}{2}}{k_i}^{-\frac{1}{2}}$$

$$= \frac{\tau}{\pi}\frac{1}{1-(1-\tau^2)k_i}(1-k_i)^{-\frac{1}{2}}{k_i}^{-\frac{1}{2}} \quad \blacksquare$$

Since $k_i \in [0,1]$, by Fubini's theorem (or law of total expectation), $E[x] = E\big[E[x|y]\big]$, posterior mean $E[\beta_i|y_i]$ is given as:

$$E[\beta_i|y_i] = E\left[E\big[\beta_i\big|y_i, {\lambda_i}^2\tau^2\big]\right] = E[(1-k_i)y_i]$$

$$= \int_0^1 (1-k_i)y_i \times \pi(k_i|y_i)dk_i \tag{A.25}$$

$$= \left(\int_0^1 \pi(k_i|y_i)dk_i - \int_0^1 k_i\pi(k_i|y_i)dk_i\right)y_i$$

$$= (1 - E[(k_i|y_i)])y_i$$

The posterior mean $E[\beta_i|y_i]$ will be referred to as the horseshoe estimator and denoted as $T_\tau(y)$,

$$T_\tau(y) = E[\beta_i|y_i] = (1 - E[(k_i|y_i)])y_i. \quad \text{(A.26)}$$

The horseshoe prior takes its name from the prior on k_i, which is given by

$$P_k(k_i\ ;\ \tau\ ) = \frac{\tau}{\pi}\frac{1}{1-(1-\tau^2)k_i}(1-k_i)^{-\frac{1}{2}}k_i^{-\frac{1}{2}}.$$

If $\tau = 1$,

$$P_k(k_i\ ;\ \tau = 1\ ) = \frac{1}{\pi}(1-k_i)^{-\frac{1}{2}}k_i^{-\frac{1}{2}} \quad \text{(A.27)}$$

which is the pdf of $Beta\left(\frac{1}{2},\frac{1}{2}\right)$ and it looks like a horseshoe.

Proof: $Beta\left(\frac{1}{2},\frac{1}{2}\right)$ has the same pdf as $P_k(k_i\ ;\ \tau = 1\ )$.

The pdf of $Beta(\alpha,\beta)$ is

$$f(x;\ \alpha,\beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha,\ \beta)}, \quad B(\alpha,\beta) = \int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx \quad ,0 \le x \le 1$$

Let $x = (sin\theta)^2$, then

$$B(\alpha,\beta) = \int_0^{\frac{\pi}{2}}(sin\theta)^{2(\alpha-1)}(1-(sin\theta)^2)^{\beta-1}2sin\theta cos\theta d\theta$$

$$= 2\int_0^{\frac{\pi}{2}}(sin\theta)^{(2\alpha-1)}(cos\theta)^{(2\beta-1)}d\theta.$$

Let $\alpha = \frac{1}{2}$, and $\beta = \frac{1}{2}$,

$$B\left(\alpha = \frac{1}{2},\beta = \frac{1}{2}\right) = 2\int_0^{\frac{\pi}{2}}d\theta = \pi.$$

Therefore,

$$f\left(x;\ \alpha = \frac{1}{2},\beta = \frac{1}{2}\right) = \frac{x^{-\frac{1}{2}}(1-x)^{-\frac{1}{2}}}{\pi}$$ ■

Detailed derivation of posterior of $(k_i|y_i)$ is shown below:

$$y_i|\theta_i \sim \mathcal{N}(\theta_i, 1)$$

$$\theta_i|\lambda_i, \tau \sim \mathcal{N}(0, {\lambda_i}^2\, \tau^2)$$

Marginal likelihood:

$$P(y_i|\lambda_i, \tau) = \int p(y_i\,|\theta_i) p(\theta_i|\lambda_i, \tau) d\theta$$

$$k_i = \frac{1}{1 + {\lambda_i}^2\, \tau^2}$$

$$P(y_i|k_i, \tau) = {k_i}^{\frac{1}{2}} exp(-k_i {y_i}^2/2).$$

Therefore, posterior of $(k_i|y_i)$ is:

$$P(k_i|y_i) \propto P(\boldsymbol{y}_i|\boldsymbol{k}_i, \boldsymbol{\tau}) \times P_k(k_i\,;\,\tau\,)$$

$$\propto {k_i}^{\frac{1}{2}} exp(-k_i {y_i}^2/2) \times \frac{\tau}{\pi} \frac{1}{1-(1-\tau^2)k_i} (1-k_i)^{-\frac{1}{2}} {k_i}^{-\frac{1}{2}}$$

A.3.3. Three Parameter Beta ($\mathcal{TPB}$) Prior

In the forthcoming text, $\Gamma(.)$ denotes the gamma function, $\mathcal{G}(\mu, \nu)$denotes a gamma distribution with shape and **rate** parameters μ and ν.

Definition 1. The three-parameter beta (TPB) distribution for a random variable X is defined by the density function,

$$f(x;, a, b, \phi) = \frac{\Gamma(\mathrm{a}+\mathrm{b})}{\Gamma(\mathrm{a})\Gamma(\mathrm{b})} \phi^b x^{b-1} (1-x)^{a-1} \{1 + (\phi - 1)x\}^{-(a+b)} \qquad \text{(A.28)}$$

for $0 < x < 1, a > 0,\ b > 0, \phi > 0$, and is denoted by $\mathcal{TPB}(a, b, \phi)$.

Recall, in horseshoe prior,

$$p_\tau(k_i) = \frac{\tau}{\pi} \frac{1}{1-(1-\tau^2)k_i} (1-k_i)^{-\frac{1}{2}} {k_i}^{-\frac{1}{2}}.$$

It's a special case of $\boldsymbol{\mathcal{TPB}}\left(x;, a = \frac{1}{2}, b = \frac{1}{2}, \phi = \tau^2\right)$.

Definition 2. The TPB normal scale mixture representation for the distribution of random variable $\theta_j$is given by

$$\theta_j|\rho_j \sim \mathcal{N}\left(0, \frac{1}{\rho_j} - 1\right), \quad \rho_j \sim \mathcal{TPB}(a, b, \phi) \tag{A.29}$$

where $> 0, b > 0, and\ \phi > 0$. The resulting marginal distribution on θ_j is denoted by $\mathcal{TPBN}(a, b, \phi)$, equivalence of three hierarchical representations.

Proposition 1. If $\theta_j \sim \mathcal{TPBN}(a, b, \phi)$, then

1) $\theta_j \sim \mathcal{N}(0, \tau_j), \tau_j \sim \mathcal{G}(a, \lambda_j)\ and\ \lambda_j \sim \mathcal{G}(b, \phi)$.

2) $\theta_j \sim \mathcal{N}(0, \tau_j), \pi(\tau_j) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \phi^{-a} {\tau_j}^{a-1} \left(1 + \frac{\tau_j}{\phi}\right)^{-(a+b)}$,

which implies that $\frac{\tau_j}{\phi} \sim \beta'(a, b)$, the inverted beta (or beta prime) distribution with parameters a and b.

Proof: $\pi(\tau_j) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \phi^{-a} {\tau_j}^{a-1} \left(1 + \frac{\tau_j}{\phi}\right)^{-(a+b)}$

$$\tau_j \sim \mathcal{G}(a, \lambda_j)$$

$$\pi(\tau_j|\lambda_j) = \frac{{\lambda_j}^a}{\Gamma(a)} {\tau_j}^{a-1} e^{-\lambda_j \tau_j}$$

$$\lambda_j \sim \mathcal{G}(b, \phi)$$

$$\pi(\lambda_j; b, \phi) = \frac{\phi^b}{\Gamma(b)} {\lambda_j}^{b-1} e^{-\phi \lambda_j}$$

Marginal of π is

$$\pi(\tau_j) = \int_0^\infty \pi(\tau_j, \lambda_j)\, d\lambda_j$$

$$= \int_0^\infty \pi(\tau_j|\lambda_j)\pi(\lambda_j)\, d\lambda_j$$

$$= \int_0^{\infty} \frac{{\lambda_j}^{a}}{\Gamma(a)} {\tau_j}^{a-1} e^{-\lambda_j \tau_j} \frac{\phi^b}{\Gamma(b)} {\lambda_j}^{b-1} e^{-\phi\lambda_j} d\lambda_j$$

$$= \frac{{\tau_j}^{a-1}\phi^b}{\Gamma(a)\Gamma(b)} \int_0^{\infty} {\lambda_j}^{(a+b)-1} e^{-(\tau_j+\phi)\lambda_j} \, d\lambda_j$$

$$= \frac{{\tau_j}^{a-1}\phi^b}{\Gamma(a)\Gamma(b)} \times \frac{1}{\tau_j+\phi} \times \frac{1}{(\tau_j+\phi)^{(a+b)-1}} \int_0^{\infty} \left[(\tau_j+\phi)\lambda_j\right]^{(a+b)-1} e^{-(\tau_j+\phi)\lambda_j} \, d(\tau_j+\phi)\lambda_j$$

$$= \frac{{\tau_j}^{a-1}\phi^b}{\Gamma(a)\Gamma(b)} \times \frac{1}{\tau_j+\phi} \times \frac{1}{(\tau_j+\phi)^{(a+b)-1}} \times \Gamma(a+b)$$

$$= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{{\tau_j}^{a-1}\phi^b}{(\tau_j+\phi)^{(a+b)}}$$

$$= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \frac{{\tau_j}^{a-1}\phi^{-a}\phi^{a+b}}{(\tau_j+\phi)^{(a+b)}}$$

$$= \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \phi^{-a} {\tau_j}^{a-1} \left(1+\frac{\tau_j}{\phi}\right)^{-(a+b)} \quad \blacksquare$$

note: $\Gamma(\alpha) = \int_0^{\infty} x^{\alpha-1} e^{-x} dx$

Proof: Proposition 1 and definition 2 have consistent results

Definition 2,

$$\theta_j|\rho_j \sim \mathcal{N}\left(0, \frac{1}{\rho_j} - 1\right), \quad \rho_j \sim \mathcal{TPB}(a, b, \phi)$$

$$f(\rho_j;, a, b, \phi) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \phi^b {\rho_j}^{b-1} (1-\rho_j)^{a-1} \{1+(\phi-1)\rho_j\}^{-(a+b)}$$

Proposition 1,

$$\theta_j \sim \mathcal{N}(0, \tau_j),$$

$$\pi(\tau_j) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} \phi^{-a} {\tau_j}^{a-1} \left(1+\frac{\tau_j}{\phi}\right)^{-(a+b)}$$

Jacobian of the transformation $\tau_j = \frac{1}{\rho_j} - 1$

$$\frac{\partial \tau_j}{\partial \rho_j} = -\frac{1}{{\rho_j}^2}$$

$$\pi\left(\frac{1}{\rho_j} - 1\right)\left|\frac{\partial \tau_j}{\partial \rho_j}\right| = \frac{\Gamma(\mathrm{a}+\mathrm{b})}{\Gamma(\mathrm{a})\Gamma(\mathrm{b})}\phi^{-a}\left(\frac{1}{\rho_j} - 1\right)^{a-1}\left(1 + \frac{\left(\frac{1}{\rho_j} - 1\right)}{\phi}\right)^{-(a+b)} \times \frac{1}{{\rho_j}^2}$$

$$= \frac{\Gamma(\mathrm{a}+\mathrm{b})}{\Gamma(\mathrm{a})\Gamma(\mathrm{b})}\phi^{-a+(a+b)}\frac{\left(1-\rho_j\right)^{(a-1)}}{{\rho_j}^{(a-1)+2}}\left(\phi + \left(\frac{1}{\rho_j} - 1\right)\right)^{-(a+b)}$$

$$= \frac{\Gamma(\mathrm{a}+\mathrm{b})}{\Gamma(\mathrm{a})\Gamma(\mathrm{b})}\phi^{b}\left(1-\rho_j\right)^{(a-1)}{\rho_j}^{b-1}\{1 + (\phi - 1)\rho_j\}^{-(a+b)}$$

$$= f(\rho_j;, a, b, \phi) \qquad \blacksquare$$

Proof: $\frac{\tau_j}{\phi} \sim \beta'(a, b)$

$$\pi(\tau_j) = \frac{\Gamma(\mathrm{a}+\mathrm{b})}{\Gamma(\mathrm{a})\Gamma(\mathrm{b})}\phi^{-a}{\tau_j}^{a-1}\left(1 + \frac{\tau_j}{\phi}\right)^{-(a+b)}$$

let $\frac{\tau_j}{\phi} = y$, then $\tau_j = y\phi$, using transformation

$$f\left(\frac{\tau_j}{\phi}\right) = f(y) = \pi(y\phi)\left|\frac{\partial \tau_j}{\partial y}\right| = \frac{\Gamma(\mathrm{a}+\mathrm{b})}{\Gamma(\mathrm{a})\Gamma(\mathrm{b})}\phi^{-a}(y\phi)^{a-1}\left(1 + \frac{y\phi}{\phi}\right)^{-(a+b)} \times \phi$$

$$= \frac{\Gamma(\mathrm{a}+\mathrm{b})}{\Gamma(\mathrm{a})\Gamma(\mathrm{b})}y^{a-1}(1+y)^{-(a+b)}$$

$$= \frac{y^{a-1}(1+y)^{-a-b}}{B(a,b)}$$

$$= \beta'(a, b) \qquad \blacksquare$$

APPENDIX B. PROOF OF PARAMETER ESTIMATES IN BICMIX

B.1. Eq. (33)

Expected complete log-likelihood for parameters related to Λ:

$$\Lambda\colon \mathbb{Q}(\Theta_\Lambda) = < \ell_c(\Theta_\Lambda, \Lambda|\mathrm{z}, \mathrm{X}, \mathrm{Y}) >$$

$$\langle \ln(p(\Theta_\Lambda, \Lambda|\mathrm{z}, \mathrm{X}, \mathrm{Y})) \rangle$$

$$\propto \langle \ln(p(Y|\,\Lambda, X, \Theta_\Lambda, \mathrm{z})) + \ln(\mathrm{p}(\Lambda|\mathrm{z}, \Theta_\Lambda) \times p(\mathrm{z}|\pi)) + \ln(p(\Theta_\Lambda) + \ln(p(\pi|\alpha, \beta)) \rangle$$

First part: $\langle \ln(p(Y|\,\Lambda, X, \Theta_\Lambda, \mathrm{z})) \rangle$

$$Y = \Lambda \mathrm{X} + \varepsilon$$

$$\varepsilon_{.,i} \sim \mathcal{N}(0, \Psi), \qquad \Psi = diag(\psi_1, \ldots, \psi_p)$$

$$Y_{.,i} \sim \mathcal{N}(\Lambda\,\mathrm{X}, \qquad \Psi)$$

$$p(Y \mid \Lambda, X, \Psi) \propto |\Psi|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(Y - \Lambda X)^T \Psi^{-1} (Y - \Lambda X)\right)$$

Expected Log-likelihood over all latent variables and parameters except parameters related to **Λ**:

$$\langle \ln\left(\prod_{j=1}^{P}\prod_{i=1}^{n} p(y_{j,i} \mid \Lambda, X)\right)\rangle \propto \langle \ln\left(\prod_{j=1}^{P}\prod_{i=1}^{n} |\psi_{j,j}|^{-\frac{1}{2}} \exp\left(-\frac{(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}x_{k,i})^2}{2\psi_{j,j}}\right)\right)\rangle$$

$$\propto \langle -\frac{n}{2}\ln(|\Psi|) - \sum_{j=1}^{P}\sum_{i=1}^{n}\frac{(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}x_{k,i})^2}{2\psi_{j,j}}\rangle$$

$$\propto -\frac{n}{2}\ln(|\Psi|) - \sum_{j=1}^{P}\sum_{i=1}^{n}\frac{\langle(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}x_{k,i})^2\rangle}{2\psi_{j,j}}$$

$$\propto -\frac{n}{2}\ln(|\Psi|) - \sum_{j=1}^{P}\sum_{i=1}^{n}\frac{\langle(y_{j,i} - \sum_{k=1}^{K}(\Lambda_{j,k}x_{k,i}))\rangle^2 + Var(y_{j,i} - \sum_{k=1}^{K}(\Lambda_{j,k}x_{k,i}))}{2\psi_{j,j}}$$

$$\propto -\frac{n}{2}\ln(|\Psi|) - \sum_{j=1}^{P}\sum_{i=1}^{n}\frac{\left(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2 + \psi_{j,j}}{2\psi_{j,j}}$$

$$\propto -\frac{n}{2}\ln(|\Psi|) - \sum_{j=1}^{P}\sum_{i=1}^{n}\frac{\left(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2}{2\psi_{j,j}}$$

<u>Second part</u>: $\langle \mathbf{ln}(\mathbf{p}(\boldsymbol{\Lambda}|\mathbf{z},\boldsymbol{\Theta}_{\boldsymbol{\Lambda}})\rangle$

$$\prod_{j=1}^{P}\prod_{k=1}^{K} p\left(\Lambda_{j,k}\,|z_k,\Theta_\Lambda\right)p(z_k|\pi) = \prod_{j=1}^{P}\prod_{k=1}^{K}\left[\pi \times p(\Lambda_{j,k}|\Theta_\Lambda)\right]^{z_k}\left[(1-\pi)\times p(\Lambda_{j,k}|\phi_k)\right]^{1-z_k}$$

$$\mathcal{N}\left(\Lambda_{j,k}\middle|\theta_{j,k}\right) \propto {\theta_{j,k}}^{-\frac{1}{2}}\exp\left(-\frac{{\Lambda_{j,k}}^2}{2\theta_{j,k}}\right)$$

$$\mathcal{G}a\left(\theta_{j,k}\,|a,\delta_{j,k}\right) \propto {\delta_{j,k}}^{a}{\theta_{j,k}}^{a-1}\exp(-\delta_{j,k}\theta_{j,k})$$

$$\mathcal{G}a\left(\delta_{j,k}\,|b,\phi_k\right) \propto {\phi_k}^{b}{\delta_{j,k}}^{a-1}\exp(-\phi_k\delta_{j,k})$$

$$\mathcal{N}\left(\Lambda_{j,k}\middle|\phi_k\right) \propto {\phi_k}^{-\frac{1}{2}}\exp\left(-\frac{{\Lambda_{j,k}}^2}{2\phi_k}\right)$$

$$\langle \ln\left(\mathrm{p}(\Lambda|\mathrm{z},\Theta_\Lambda)p(\mathrm{z}|\Theta_\Lambda)\right)\rangle$$

$$= \sum_{j=1}^{P}\sum_{k=1}^{K}\left\{\langle z_k\ln(\pi) + z_k\ln\left(p\left(\Lambda_{j,k}\middle|\Theta_\Lambda\right)\right) + (1-z_k)\ln(1-\pi) + (1-z_k)\ln\left(p(\Lambda_{j,k}|\phi_k)\right)\rangle\right\}$$

$$\propto \sum_{j=1}^{P}\sum_{k=1}^{K}\{\langle z_k\rangle\ln(\pi) + (1-\langle z_k\rangle)\ln(1-\pi)\}$$

$$+ \sum_{j=1}^{P}\sum_{k=1}^{K}\langle z_k\rangle\left\{-\frac{1}{2}\ln\left(\theta_{j,k}\right) - \frac{{\Lambda_{j,k}}^2}{2\theta_{j,k}} + a\ln\left(\delta_{j,k}\right) + (a-1)\ln\left(\theta_{j,k}\right) - \delta_{j,k}\theta_{j,k} + b\ln(\phi_k)\right.$$

$$\left. + (b-1)\ln\left(\delta_{j,k}\right) - \phi_k\delta_{j,k}\right\}$$

$$+ \sum_{j=1}^{P}\sum_{k=1}^{K}(1-\langle z_k\rangle)\left\{-\frac{1}{2}\ln(\phi_k) - \frac{{\Lambda_{j,k}}^2}{2\phi_k}\right\}$$

Third Part: $\langle \ln\,(p(\Theta_\Lambda)\rangle$

$$p(\Theta_\Lambda) = \left[\prod_{k=1}^{K} \mathcal{G}a(\phi_k\,|c,\tau_k)\mathcal{G}a(\,\tau_k\,|d,\eta)\right] \mathcal{G}a(\,\eta\,|e,\gamma)\mathcal{G}a(\,\gamma\,|f,\nu)$$

$$\mathcal{G}a(\phi_k\,|c,\tau_k) \;\propto\; {\tau_k}^{c}{\phi_k}^{c-1}\exp(-\tau_k\phi_k)$$

$$\mathcal{G}a(\,\tau_k\,|d,\eta) \propto\; \eta^{d}{\tau_k}^{d-1}\exp(-\,\eta\tau_k)$$

$$\mathcal{G}a(\,\eta\,|e,\gamma) \propto\; \gamma^{e}\eta^{e-1}\exp(-\gamma\eta)$$

$$\mathcal{G}a(\,\gamma|f,\nu) \propto\; \nu^{f}\gamma^{f-1}\exp(-\nu\gamma)$$

$$\langle \ln\big(p(\Theta_\Lambda)\big)\rangle \;\propto \sum_{k=1}^{K}\{c\ln(\tau_k) + (c-1)\ln(\phi_k) - \tau_k\phi_k + d\ln(\eta) + (d-1)\ln(\tau_k) - \eta\tau_k\}$$

$$+e\ln(\gamma) + (e-1)\ln(\eta) - \gamma\eta + f\ln(\nu) + (f-1)\ln(\gamma) - \nu\gamma$$

Fourth Part: $\langle \mathbf{ln}(\boldsymbol{p}(\boldsymbol{\pi}|\boldsymbol{\alpha},\boldsymbol{\beta})\rangle$

$$\mathcal{B}eta(\pi|\alpha,\beta) \propto\; \pi^{\alpha-1}(1-\pi)^{\beta-1}$$

$$\langle \ln(p(\pi|\alpha,\beta)\rangle \;\propto (\alpha-1)\ln(\pi) + (\beta-1)\ln(1-\pi)$$

therefore,

Equation (33) = Part one + part two + part three + part four

$$\mathbb{Q}(\Theta_\Lambda) \;\propto\; \langle \ln(p(Y|\,\Lambda, X, \Theta_\Lambda, \mathrm{z})) + \ln(\mathrm{p}(\Lambda|\mathrm{z},\Theta_\Lambda)\times p(\mathrm{z}|\pi)) + \;\ln\,(p(\Theta_\Lambda) + \mathrm{l\,n}(p(\pi|\alpha,\beta))\rangle$$

$$\propto -\frac{n}{2}\ln(|\Psi|) - \sum_{j=1}^{P}\sum_{i=1}^{n}\frac{\left(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2}{2\psi_{j,j}}$$

$$+\sum_{k=1}^{K}\{\,\langle z_k\rangle\ln(\pi) + (1-\langle z_k\rangle)\ln(1-\pi)\}$$

$$+\sum_{j=1}^{P}\sum_{k=1}^{K}\langle z_k\rangle\left\{-\frac{1}{2}\ln(\theta_{j,k})-\frac{{\Lambda_{j,k}}^2}{2\theta_{j,k}}+a\ln(\delta_{j,k})+(a-1)\ln(\theta_{j,k})-\delta_{j,k}\theta_{j,k}+b\ln(\phi_k)\right.$$

$$\left.+(b-1)\ln(\delta_{j,k})-\ \phi_k\delta_{j,k}\right\}$$

$$+\sum_{j=1}^{P}\sum_{k=1}^{K}(1-\langle z_k\rangle)\left\{-\frac{1}{2}\ln(\phi_k)-\frac{{\Lambda_{j,k}}^2}{2\phi_k}\right\}$$

$$+\sum_{k=1}^{K}\{c\ln(\tau_k)+(c-1)\ln(\phi_k)-\tau_k\phi_k+d\ln(\eta)+(d-1)\ln(\tau_k)-\eta\tau_k\}$$

$$+e\ln(\gamma)+(e-1)\ln(\eta)-\gamma\eta+f\ln(\nu)+(f-1)\ln(\gamma)-\nu\gamma$$

$$+\ (\alpha-1)\ln(\pi)+(\beta-1)\ln(1-\pi)$$ ■

Computing the Maximum of A Posterior (MAP) estimates for the parameters that encourage sparsity in the **Λ** matrix,

$$\widehat{\Theta_\Lambda}=argmax_{\Theta_\Lambda}\,\mathbb{Q}(\Theta_\Lambda)$$

$$\frac{\partial\mathbb{Q}(\Theta_\Lambda)}{\partial\Theta_\Lambda}=0$$

B.2. Eq. (35), estimate of $\Lambda_{j,k}$ and its matrix form $\Lambda_{j,\cdot}$

Components with $\Lambda_{j,k}$ in $\mathbb{Q}(\Theta_\Lambda)$:

$$-\sum_{j=1}^{P}\sum_{i=1}^{n}\frac{\left(y_{j,i}-\sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2}{2\psi_{j,j}}$$

$$+\sum_{j=1}^{P}\sum_{k=1}^{K}\langle z_k\rangle\left\{-\frac{{\Lambda_{j,k}}^2}{2\theta_{j,k}}\right\}+\sum_{j=1}^{P}\sum_{k=1}^{K}(1-\langle z_k\rangle)\left\{-\frac{1}{2}\ln(\phi_k)-\frac{{\Lambda_{j,k}}^2}{2\phi_k}\right\}$$

$$\frac{\partial \mathbb{Q}(\boldsymbol{\Theta}_{\boldsymbol{\Lambda}})}{\partial \boldsymbol{\Lambda}} = -\sum_{\mathrm{j}=1}^{\mathrm{P}}\sum_{i=1}^{n}\left\{\frac{2(\mathrm{y}_{\mathrm{j,i}} - \sum_{\mathrm{k}'=1}^{\mathrm{K}}\Lambda_{\mathrm{j,k}'}\langle \mathrm{x}_{\mathrm{k}',\mathrm{i}}\rangle)}{2\psi_{\mathrm{j,j}}}\left(-\sum_{k=1}^{K}\langle \mathrm{x}_{\mathrm{k,i}}\rangle\right)\right\} + \sum_{\mathrm{j}=1}^{\mathrm{P}}\sum_{\mathrm{k}=1}^{\mathrm{K}}\langle \mathrm{z}_{\mathrm{k}}\rangle\left\{-\frac{2\Lambda_{\mathrm{j,k}}}{2\theta_{\mathrm{j,k}}}\right\}$$

$$+\sum_{\mathrm{j}=1}^{\mathrm{P}}\sum_{\mathrm{k}=1}^{\mathrm{K}}(1-\langle \mathrm{z}_{\mathrm{k}}\rangle)\left\{-\frac{2\Lambda_{\mathrm{j,k}}}{2\phi_{\mathrm{k}}}\right\}$$

$$= \sum_{j=1}^{P}\sum_{k=1}^{K}\left\{\psi_{jj}^{-1}\sum_{i=1}^{n}\left[(\mathrm{y}_{\mathrm{j,i}}\langle \mathrm{x}_{\mathrm{k,i}}\rangle - \sum_{k'\neq k}\Lambda_{\mathrm{j,k}'}\langle \mathrm{x}_{\mathrm{k}',\mathrm{i}}\mathrm{x}_{\mathrm{k,i}}\rangle - \Lambda_{j,k}\langle x_{k,i}^{2}\rangle)\right] - \Lambda_{\mathrm{j,k}}\frac{\langle \mathrm{z}_{\mathrm{k}}\rangle}{\theta_{\mathrm{j,k}}}\right.$$

$$\left. - \Lambda_{\mathrm{j,k}}\frac{(1-\langle \mathrm{z}_{\mathrm{k}}\rangle)}{\phi_{\mathrm{k}}}\right\} = 0$$

$$\widehat{\Lambda_{j,k}} = \frac{\psi_{jj}^{-1}\sum_{i=1}^{n}\left[(\mathrm{y}_{\mathrm{j,i}}\langle \mathrm{x}_{\mathrm{k,i}}\rangle - \sum_{k'\neq k}\Lambda_{\mathrm{j,k}'}\langle \mathrm{x}_{\mathrm{k}',\mathrm{i}}\mathrm{x}_{\mathrm{k,i}}\rangle)\right]}{\psi_{jj}^{-1}\sum_{i=1}^{n}\langle x_{k,i}^{2}\rangle + \frac{\langle \mathrm{z}_{\mathrm{k}}\rangle}{\theta_{\mathrm{j,k}}} + \frac{(1-\langle \mathrm{z}_{\mathrm{k}}\rangle)}{\phi_{\mathrm{k}}}}$$

Note: $\langle x_{k',i}\rangle\langle x_{k,i}\rangle = \langle x_{k',i}x_{k,i}\rangle$ because factors are independent,

$$E[xy] = E[x]E[y] + cov(x,y) = E[x]E[y]$$

Its matrix form is given as

$$\frac{\partial \mathbb{Q}(\boldsymbol{\Theta}_{\boldsymbol{\Lambda}})}{\partial \boldsymbol{\Lambda}} = -\sum_{\mathrm{j}=1}^{\mathrm{P}}\sum_{i=1}^{n}\left\{\frac{2(\mathrm{y}_{\mathrm{j,i}} - \sum_{\mathrm{k}'=1}^{\mathrm{K}}\Lambda_{\mathrm{j,k}'}\langle \mathrm{x}_{\mathrm{k}',\mathrm{i}}\rangle)}{2\psi_{\mathrm{j,j}}}\left(-\sum_{k=1}^{K}\langle \mathrm{x}_{\mathrm{k,i}}\rangle\right)\right\} + \sum_{\mathrm{j}=1}^{\mathrm{P}}\sum_{\mathrm{k}=1}^{\mathrm{K}}\langle \mathrm{z}_{\mathrm{k}}\rangle\left\{-\frac{2\Lambda_{\mathrm{j,k}}}{2\theta_{\mathrm{j,k}}}\right\}$$

$$+\sum_{\mathrm{j}=1}^{\mathrm{P}}\sum_{\mathrm{k}=1}^{\mathrm{K}}(1-\langle \mathrm{z}_{\mathrm{k}}\rangle)\left\{-\frac{2\Lambda_{\mathrm{j,k}}}{2\phi_{\mathrm{k}}}\right\}$$

$$= \sum_{j=1}^{P}\left\{\ \left[(\ \psi_{jj}^{-1}\mathrm{y}_{\mathrm{j},\cdot}\langle X\rangle^{T} - \psi_{jj}^{-1}\Lambda_{\mathrm{j},\cdot}\langle XX^{T}\rangle\right] - \frac{\langle Z\rangle}{\Theta_{j,\cdot}}\Lambda_{\mathrm{j},\cdot}\frac{(1-\langle Z\rangle)}{\Phi}\Lambda_{\mathrm{j},\cdot}\right\} = 0$$

Note: $\sum_{i=1}^{n}\sum_{\mathrm{k}'=1}^{\mathrm{K}}\Lambda_{\mathrm{j,k}'}\langle \mathrm{x}_{\mathrm{k}',\mathrm{i}}\rangle\left(\sum_{k=1}^{K}\langle \mathrm{x}_{\mathrm{k,i}}\rangle\right) = \Lambda_{\mathrm{j},\cdot}\langle XX^{T}\rangle.$

$$\widehat{\Lambda_{j,\cdot}} = \mathrm{y}_{\mathrm{j},\cdot}\psi_{jj}^{-1}\langle X\rangle^{T}\left(\langle X\psi_{jj}^{-1}X^{T}\rangle + \langle Z\rangle\Theta_{j,\cdot}{}^{-1} + (1-\langle Z\rangle)\Phi^{-1}\right)^{-1} \qquad \blacksquare$$

where

$$\Theta_{j,\cdot} = \begin{pmatrix} \theta_{j,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \theta_{j,K} \end{pmatrix} \quad \Phi = \begin{pmatrix} \phi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \phi_K \end{pmatrix} \quad \langle Z\rangle = \begin{pmatrix} \langle \mathrm{z}_1\rangle & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \langle \mathrm{z}_\mathrm{K}\rangle \end{pmatrix}$$

B.3. Eq. (38), estimate of $\langle \mathrm{x}_{\mathrm{k,i}}\rangle$

$$\frac{\partial \mathbb{Q}(\Theta_\Lambda)}{\partial\langle X\rangle} = 0$$

Components related to $\langle \mathrm{x}_{\mathrm{k,i}}\rangle$ in $\mathbb{Q}(\Theta_\mathrm{X})$:

$$-\sum_{\mathrm{j}=1}^{\mathrm{P}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\frac{\left(\mathrm{y}_{\mathrm{j,i}}-\sum_{\mathrm{k}=1}^{\mathrm{K}}\Lambda_{\mathrm{j,k}}\langle \mathrm{x}_{\mathrm{k,i}}\rangle\right)^{2}}{2\psi_{\mathrm{j,j}}} + \sum_{\mathrm{k}=1}^{\mathrm{K}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\langle \mathrm{o}_\mathrm{k}\rangle\left\{-\frac{\langle \mathrm{x}_{\mathrm{k,i}}^2\rangle}{2\sigma_{\mathrm{k,i}}}\right\} + \sum_{\mathrm{k}=1}^{\mathrm{K}}\sum_{\mathrm{i}=1}^{\mathrm{n}}(1-\langle \mathrm{o}_\mathrm{k}\rangle)\left\{-\frac{\langle \mathrm{x}_{\mathrm{k,i}}^2\rangle}{2\omega_\mathrm{k}}\right\}$$

$$\propto -\sum_{\mathrm{j}=1}^{\mathrm{P}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\frac{\left(\mathrm{y}_{\mathrm{j,i}}-\sum_{\mathrm{k}=1}^{\mathrm{K}}\Lambda_{\mathrm{j,k}}\langle \mathrm{x}_{\mathrm{k,i}}\rangle\right)^{2}}{2\psi_{\mathrm{j,j}}} + \sum_{\mathrm{k}=1}^{\mathrm{K}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\langle \mathrm{o}_\mathrm{k}\rangle\left\{-\frac{\langle \mathrm{x}_{\mathrm{k,i}}\rangle^2}{2\sigma_{\mathrm{k,i}}}\right\} + \sum_{\mathrm{k}=1}^{\mathrm{K}}\sum_{\mathrm{i}=1}^{\mathrm{n}}(1-\langle \mathrm{o}_\mathrm{k}\rangle)\left\{-\frac{\langle \mathrm{x}_{\mathrm{k,i}}\rangle^2}{2\omega_\mathrm{k}}\right\}$$

Note: $\langle \mathrm{x}_{\mathrm{k,i}}^2\rangle = \langle \mathrm{x}_{\mathrm{k,i}}\rangle^2 + var(\mathrm{x}_{\mathrm{k,i}})$

$$\frac{\partial \mathbb{Q}(\Theta_\Lambda)}{\partial\langle X\rangle} = -\sum_{\mathrm{j}=1}^{\mathrm{P}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\frac{2\left(\mathrm{y}_{\mathrm{j,i}} - \sum_{\mathrm{k}'=1}^{\mathrm{K}}\Lambda_{\mathrm{j,k}'}\langle \mathrm{x}_{\mathrm{k}',\mathrm{i}}\rangle\right)}{2\psi_{\mathrm{j,j}}}\left(-\sum_{k=1}^{K}\Lambda_{\mathrm{j,k}}\right) + \sum_{\mathrm{k}=1}^{\mathrm{K}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\langle \mathrm{o}_\mathrm{k}\rangle\left\{-\frac{2\langle \mathrm{x}_{\mathrm{k,i}}\rangle}{2\sigma_{\mathrm{k,i}}}\right\}$$

$$+\sum_{\mathrm{k}=1}^{\mathrm{K}}\sum_{\mathrm{i}=1}^{\mathrm{n}}(1-\langle \mathrm{o}_\mathrm{k}\rangle)\left\{-\frac{2\langle \mathrm{x}_{\mathrm{k,i}}\rangle}{2\omega_\mathrm{k}}\right\}$$

$$= -\sum_{\mathrm{j}=1}^{\mathrm{P}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\sum_{k=1}^{K}\left(\mathrm{y}_{\mathrm{j,i}}\psi_{\mathrm{j,j}}{}^{-1}\Lambda_{\mathrm{j,k}} - \psi_{\mathrm{j,j}}{}^{-1}\sum_{\mathrm{k}'=1}^{\mathrm{K}}\Lambda_{\mathrm{j,k}'}\langle \mathrm{x}_{\mathrm{k}',\mathrm{i}}\rangle\Lambda_{\mathrm{j,k}}\right) + \sum_{\mathrm{k}=1}^{\mathrm{K}}\sum_{\mathrm{i}=1}^{\mathrm{n}}\langle \mathrm{o}_\mathrm{k}\rangle\sigma_{\mathrm{k,i}}{}^{-1}\langle \mathrm{x}_{\mathrm{k,i}}\rangle$$

$$+\sum_{\mathrm{k}=1}^{\mathrm{K}}\sum_{\mathrm{i}=1}^{\mathrm{n}}(1-\langle \mathrm{o}_\mathrm{k}\rangle)\omega_\mathrm{k}{}^{-1}\langle \mathrm{x}_{\mathrm{k,i}}\rangle$$

$$= \sum_{i=1}^{n} \left\{ \Lambda^T \Psi^{-1} Y_{\cdot,i} - \Lambda^T \Psi^{-1} \Lambda \langle X_{\cdot,i} \rangle - (\mathrm{I} - \langle \mathrm{O} \rangle) \Omega^{-1} \langle X_{\cdot,i} \rangle - \langle \mathrm{O} \rangle \Sigma_i^{-1} \langle X_{\cdot,i} \rangle \right\} = 0$$

$$\langle X_{\cdot,i} \rangle = (\Lambda^T \Psi^{-1} \Lambda + \langle \mathrm{O} \rangle \Sigma_i^{-1} + (\mathrm{I} - \langle \mathrm{O} \rangle) \Omega^{-1})^{-1} \Lambda^T \Psi^{-1} Y_{\cdot,i} \qquad (38) \blacksquare$$

where

$$\Sigma_i = \begin{pmatrix} \sigma_{1,i} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_{K,i} \end{pmatrix} \quad \Omega = \begin{pmatrix} \omega_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \omega_K \end{pmatrix} \quad \langle \mathrm{O} \rangle = \begin{pmatrix} \langle \mathrm{o}_1 \rangle & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \langle \mathrm{o}_\mathrm{K} \rangle \end{pmatrix}$$

B.4. Eq. (42), estimate of $\boldsymbol{\theta_{j,k}}$

Components having $\theta_{j,k}$ in $\mathbb{Q}(\Theta_\Lambda)$:

$$\sum_{j=1}^{P} \sum_{k=1}^{K} \langle z_k \rangle \left\{ -\frac{1}{2} \ln(\theta_{j,k}) - \frac{{\Lambda_{j,k}}^2}{2\theta_{j,k}} + (a-1) \ln(\theta_{j,k}) - \delta_{j,k} \theta_{j,k} \right\}$$

$$\frac{\partial \mathbb{Q}(\Theta_\Lambda)}{\partial \theta_{j,k}} = \langle z_k \rangle \left\{ -\frac{1}{2\theta_{j,k}} + \frac{{\Lambda_{j,k}}^2}{2{\theta_{j,k}}^2} + \frac{(a-1)}{\theta_{j,k}} - \delta_{j,k} \right\} = 0$$

$$\delta_{j,k} {\theta_{j,k}}^2 - \left(a - 1 - \frac{1}{2} \right) \theta_{j,k} - \frac{1}{2} {\Lambda_{j,k}}^2 = 0$$

$$\widehat{\theta_{\mathrm{j,k}}} = \frac{(a - \frac{3}{2}) \pm \sqrt{(a - \frac{3}{2})^2 - 4\delta_{j,k}(-\frac{1}{2}{\Lambda_{j,k}}^2)}}{2\delta_{j,k}}$$

$$= \frac{(2a-3) + \sqrt{(2a-3)^2 + 8{\Lambda_{j,k}}^2 \delta_{j,k}}}{4\delta_{j,k}} \qquad \blacksquare$$

B.5. Eq. (44), estimate of $\boldsymbol{\delta_{j,k}}$

Components having $\delta_{\mathrm{j,k}}$ in $\mathbb{Q}(\Theta_\Lambda)$:

$$\sum_{j=1}^{P} \sum_{k=1}^{K} \langle z_k \rangle \{ a \ln(\delta_{j,k}) - \delta_{j,k} \theta_{j,k} + (b-1) \ln(\delta_{j,k}) - \phi_k \delta_{j,k} \}$$

$$\frac{\partial \mathbb{Q}(\Theta_\Lambda)}{\partial \delta_{j,k}} = \langle z_k \rangle \left\{ \frac{a}{\delta_{j,k}} - \theta_{j,k} + \frac{b-1}{\delta_{j,k}} - \phi_k \right\} = 0$$

$$\widehat{\delta_{\mathrm{j,k}}} = \frac{a+b-1}{\theta_{j,k}+\phi_k} \qquad \blacksquare$$

B.6. Eq. (54), estimate of $\boldsymbol{\tau_k}$

Components having τ_k in $\mathbb{Q}(\Theta_\Lambda)$:

$$\sum_{k=1}^{K} \{c \ln(\tau_k) - \tau_k \phi_k + (d-1)\ln(\tau_k) - \eta \tau_k\}$$

$$\frac{\partial \mathbb{Q}(\Theta_\Lambda)}{\partial \tau_k} = \frac{c}{\tau_k} - \phi_k + \frac{d-1}{\tau_k} - \eta = 0$$

$$\widehat{\tau_k} = \frac{c+d-1}{\phi_k+\eta} \qquad \blacksquare$$

B.7. Eq. (55), estimate of $\boldsymbol{\eta}$

Components having η in $\mathbb{Q}(\Theta_\Lambda)$:

$$\sum_{k=1}^{K} \{d \ln(\eta) - \eta \tau_k\} + (e-1)\ln(\eta) - \gamma\eta$$

$$\frac{\partial \mathbb{Q}(\Theta_\Lambda)}{\partial \eta} = \sum_{k=1}^{K} \left(\frac{d}{\eta} - \tau_k\right) + \frac{e-1}{\eta} - \gamma = \frac{Kd}{\eta} - \sum_{k=1}^{K} \tau_k + \frac{e-1}{\eta} - \gamma = 0$$

$$\hat{\eta} = \frac{Kd+e-1}{\gamma + \sum_{k=1}^{K} \tau_k} \qquad \blacksquare$$

B.8. Eq. (56), estimate of $\boldsymbol{\gamma}$

Components having γ in $\mathbb{Q}(\Theta_\Lambda)$:

$$e \ln(\gamma) - \gamma\eta + (f-1)\ln(\gamma) - \nu\gamma$$

$$\frac{\partial \mathbb{Q}(\Theta_\Lambda)}{\partial \gamma} = \frac{e}{\gamma} - \eta + \frac{f-1}{\gamma} - \nu = 0$$

$$\hat{\gamma} = \frac{e+f-1}{\eta+\nu} \qquad \blacksquare$$

B.9. Eq. (46), estimate of $\boldsymbol{\phi_k}$

Components having ϕ_k in $\mathbb{Q}(\Theta_\Lambda)$:

$$\sum_{j=1}^{P}\sum_{k=1}^{K}\langle z_k\rangle\{b\ln(\phi_k) - \phi_k\delta_{j,k}\}$$

$$+ \sum_{j=1}^{P}\sum_{k=1}^{K}(1-\langle z_k\rangle)\left\{-\frac{1}{2}\ln(\phi_k) - \frac{{\Lambda_{j,k}}^2}{2\phi_k}\right\} + \sum_{k=1}^{K}\{(c-1)\ln(\phi_k) - \tau_k\phi_k\}$$

$$\frac{\partial \mathbb{Q}(\Theta_\Lambda)}{\partial \phi_k} = \sum_{j=1}^{P}\langle z_k\rangle\left\{\frac{b}{\phi_k} - \delta_{j,k}\right\} + \sum_{j=1}^{P}(1-\langle z_k\rangle)\left\{-\frac{1}{2\phi_k} + \frac{{\Lambda_{j,k}}^2}{2{\phi_k}^2}\right\} + \frac{c-1}{\phi_k} - \tau_k$$

$$= \frac{\langle z_k\rangle bP}{\phi_k} - \langle z_k\rangle\sum_{j=1}^{P}\delta_{j,k} - \frac{(1-\langle z_k\rangle)P}{2\phi_k} + \frac{(1-\langle z_k\rangle)\sum_{j=1}^{P}{\Lambda_{j,k}}^2}{2{\phi_k}^2} + \frac{c-1}{\phi_k} - \tau_k \ = 0$$

$$\langle z_k\rangle bP\,\phi_k - \left(\langle z_k\rangle\sum_{j=1}^{P}\delta_{j,k}\right){\phi_k}^2 - \frac{(1-\langle z_k\rangle)P}{2}\phi_k + \frac{(1-\langle z_k\rangle)\sum_{j=1}^{P}{\Lambda_{j,k}}^2}{2} + (c-1)\phi_k - \tau_k{\phi_k}^2 = 0$$

$$\left(\langle z_k\rangle\sum_{j=1}^{P}\delta_{j,k} + \tau_k\right){\phi_k}^2 - \left(\langle z_k\rangle bP - \frac{(1-\langle z_k\rangle)P}{2} + (c-1)\right)\phi_k - \frac{(1-\langle z_k\rangle)\sum_{j=1}^{P}{\Lambda_{j,k}}^2}{2}$$

$$= 0$$

$$\widehat{\phi_k} = \frac{\left(\langle z_k\rangle bP - \frac{(1-\langle z_k\rangle)P}{2} + (c-1)\right) \pm \sqrt{\left(\langle z_k\rangle bP - \frac{(1-\langle z_k\rangle)P}{2} + (c-1)\right)^2 - 4\left(\langle z_k\rangle\sum_{j=1}^{P}\delta_{j,k} + \tau_k\right)\left(-\frac{(1-\langle z_k\rangle)\sum_{j=1}^{P}{\Lambda_{j,k}}^2}{2}\right)}}{2\left(\langle z_k\rangle\sum_{j=1}^{P}\delta_{j,k} + \tau_k\right)}$$

$$= \frac{\left(\langle z_k\rangle bP - \frac{(1-\langle z_k\rangle)P}{2} + (c-1)\right) + \sqrt{\left(\langle z_k\rangle bP - \frac{(1-\langle z_k\rangle)P}{2} + (c-1)\right)^2 + 2\left(\langle z_k\rangle\sum_{j=1}^{P}\delta_{j,k} + \tau_k\right)\left((1-\langle z_k\rangle)\sum_{j=1}^{P}{\Lambda_{j,k}}^2\right)}}{2\left(\langle z_k\rangle\sum_{j=1}^{P}\delta_{j,k} + \tau_k\right)}$$

$$= \frac{H + \sqrt{H^2 + MT}}{M} \qquad \blacksquare$$

where

$$M = 2\left(\langle z_k\rangle \sum_{j=1}^{P} \delta_{j,k} + \tau_k\right), \qquad H = \left(\langle z_k\rangle bP - \frac{(1-\langle z_k\rangle)P}{2} + (c-1)\right),$$

$$T = \left((1-\langle z_k\rangle)\sum_{j=1}^{P} {\Lambda_{j,k}}^2\right)$$

B.10. Eq. (60), estimate of $\langle \boldsymbol{z_k}|\boldsymbol{\Theta_\Lambda}\rangle$

$$\langle z_k|\Theta_\Lambda\rangle = 0 \times p(z_k = 0|\Theta_\Lambda) + 1 \times p(z_k = 1|\Theta_\Lambda) = p(z_k = 1|\Theta_\Lambda)$$

$$\langle z_k|\Theta_\Lambda\rangle = p(z_k = 1|\Theta_\Lambda)$$

$$= \frac{p(z_k = 1)p(\Theta_\Lambda|z_k = 1)}{p(z_k = 0)p(\Theta_\Lambda|z_k = 0) + p(z_k = 1)p(\Theta_\Lambda|z_k = 1)}$$

$$= \frac{\pi \prod_{j=1}^{P} \mathcal{N}\left(\Lambda_{j,k}|\theta_{j,k}\right)\mathcal{G}a\left(\theta_{j,k}\,|a,\delta_{j,k}\right)\mathcal{G}a\left(\delta_{j,k}\,|b,\phi_k\right)}{(1-\pi)\mathcal{N}\left(\Lambda_{j,k}|\phi_k\right) + \pi \prod_{j=1}^{P} \mathcal{N}\left(\Lambda_{j,k}|\theta_{j,k}\right)\mathcal{G}a\left(\theta_{j,k}\,|a,\delta_{j,k}\right)\mathcal{G}a\left(\delta_{j,k}\,|b,\phi_k\right)}$$

B.11. Eq. (64), estimate of $\boldsymbol{\Psi}$

$$Y = \Lambda \mathrm{X} + \varepsilon$$

$$\varepsilon_{.,i} \sim \mathcal{N}(0, \Psi), \qquad \Psi = diag(\psi_1, \ldots, \psi_p)$$

$$\mathrm{Y} \sim \mathcal{N}(\Lambda\,\mathrm{X}, \Psi)$$

$$\langle p\left(y_{j,i} \mid \Lambda, X, \Psi\right)\rangle \propto \left|\psi_{j,j}\right|^{-\frac{1}{2}} \exp\left(-\frac{\left(y_{j,i} - \sum_{k=1}^{K} \Lambda_{j,k}\langle x_{k,i}\rangle\right)^2}{2\psi_{j,j}}\right.$$

Assuming that the residual precision has a conjugate (gamma) prior, $\frac{1}{\psi_{j,j}} \sim \mathcal{G}a(1,1)$, then

$$p\left(\frac{1}{\psi_{j,j}}\right) \propto {\frac{1}{\psi_{j,j}}}^{1-1} \exp\left(-1 \times \frac{1}{\psi_{j,j}}\right) = \exp(-{\psi_{j,j}}^{-1}).$$

Posterior of $\boldsymbol{{\psi_{j,j}}^{-1}} \mid \boldsymbol{Y}$ is

$$p\left({\psi_{j,j}}^{-1}\,|Y\right) \propto p\left({\psi_{j,j}}^{-1}\right)p(y_{j,i}|\,{\psi_{j,j}}^{-1})$$

$$\propto \left({\psi_{j,j}}^{-1}\right)^{\frac{n}{2}} exp\left\{-{\psi_{j,j}}^{-1} - \frac{\left(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2}{2\psi_{j,j}}\right\}$$

$$\propto \left({\psi_{j,j}}^{-1}\right)^{\left(\frac{n}{2}+1\right)-1} \exp\left\{-\left[1+\frac{1}{2}\left(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2\right]{\psi_{j,j}}^{-1}\right\}$$

$$\psi_{j,j}{}^{-1} \mid Y \sim \mathcal{G}a\left(\frac{n}{2}+1, 1+\frac{1}{2}\left(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2\right)$$

$$ln\left(\, p\left({\psi_{j,j}}^{-1}\,|Y\right)\right) \propto \frac{n}{2}\ln\left({\psi_{j,j}}^{-1}\right) - \left[1+\frac{1}{2}\left(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2\right]{\psi_{j,j}}^{-1}$$

$$\frac{\partial ln\left(\, p\left({\psi_{j,j}}^{-1}\,|Y\right)\right)}{\partial {\psi_{j,j}}^{-1}} = \frac{n}{2{\psi_{j,j}}^{-1}} - \left[1+\frac{1}{2}\left(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2\right] = 0$$

$$\widehat{\psi_{j,j}} = \frac{1+\frac{1}{2}\left(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2}{n/2} = \frac{\left(y_{j,i} - \sum_{k=1}^{K}\Lambda_{j,k}\langle x_{k,i}\rangle\right)^2 + 2}{n}$$

Its matrix form is

$$\widehat{\Psi} = \frac{(Y - \Lambda\langle X\rangle)(Y - \Lambda\langle X\rangle)^T + 2\mathrm{I}}{n} = \frac{YY^T - 2Y\langle X^T\rangle\Lambda^T + \Lambda\langle X\rangle\langle X^T\rangle\Lambda^T + 2\mathrm{I}}{n} \quad \blacksquare$$

B.12. Eq. (61), estimate of $\langle \ln(\boldsymbol{\pi})\rangle$

Posterior of $\pi\,|z_k$ is

$$p(\pi\,|Z) \propto Beta(\pi|\alpha,\beta)\prod_{k=1}^{K} Bern(z_k|\pi)$$

$$\propto \left(\pi^{(\alpha-1)}(1-\pi)^{(\beta-1)}\right)\prod_{k=1}^{K}\left(\pi^{z_k}(1-\pi)^{(1-z_k)}\right)$$

$$\propto \pi^{\left(\alpha+\sum_{k=1}^{K}z_k\right)-1}(1-\pi)^{\left(\beta+K-\sum_{k=1}^{K}z_k\right)-1}.$$

It shows that $\pi\,|z_k$follows a beta distribution,

$$\pi\,|z_k \sim \mathcal{B}eta(\pi|\alpha + \textstyle\sum_{k=1}^{K} z_k\,,\ \beta + K - \sum_{k=1}^{K} z_k\,).$$

Therefore,

$$\langle \ln(\pi) \rangle = \psi\left(\alpha + \sum_{k=1}^{K} z_k\right) - \psi(K + \alpha + \beta) \qquad \blacksquare$$

where ψ is the digamma function.

Note, if $x \sim \mathcal{B}eta\,(\alpha, \beta)$, then

$$\langle \ln(x) \rangle = \psi(\alpha) - \psi(\alpha + \beta),$$

where

$\psi(\cdot)$ is the digamma function

$$\psi(\alpha) \equiv \frac{d}{d\alpha} \ln \Gamma(\alpha)$$

APPENDIX C. R CODE FOR RECOVERY & RELEVANCE SCORE

```
#-----------------------------------------------------------------------------------------------#
#  This function is used to get indexes of non-zero elements in loadings and factors.
#  These elements are supposed to be continuous.
#-----------------------------------------------------------------------------------------------#
nonzero_indexes=function(sparse_vector){

 INDEXES=NULL
 for (i in 1:length(sparse_vector)){
  if (sparse_vector[i]!=0){
   first=i
   break
  }
 }
 for(i in 1:length(sparse_vector)){
  if (sparse_vector[length(sparse_vector)-(i-1)]!=0){
   last=length(sparse_vector)-(i-1)
   break
  }
 }
 if (last<first){
  print("something went wrong!!!")
  txt=paste("(first=",first,"last=", last, ")")
  print(txt)
 }
 INDEXES=c(first,last)
 return (INDEXES)
}
#-----------------------------------------------------------------------------#
#   A sparse loading and a sparse factor construct a bicluster.
#-----------------------------------------------------------------------------#
```

```
get_clusters=function(lam.est, x.est,z.est, o.est ){

  # #add up all sparse loadings and get one sparse vector
  # sparse_vector=rowSums(lam.sparse)

  k=ncol(lam.est)
  #or k=nrow(x.est)
  CLUSTERS=list()
  ind=1
  for (i in 1:k){
    #both loading and factor are sparse
    if(z.est[i]==1 & o.est[i]==1){
      LOADING_INDEXES=nonzero_indexes(lam.est[,i])
      FACTOR_INDEXES =nonzero_indexes(x.est[i,])
      CLUSTERS[[ind]]=list(LOADING_INDEXES, FACTOR_INDEXES)
      ind=ind+1
    }
  }
  return (CLUSTERS)
}
#R&R scores
One_Run_RR_Score=function(true_clusters, est_clusters){
  #all true clusters
  TRUE_CLUSTERS=list()
  for (i in 1:length(true_clusters)){
    TRUE_CLUSTERS[[i]]=c(true_clusters[[i]][[1]][1]:true_clusters[[i]][[1]][2])
  }
  #all est clusters
  EST_CLUSTERS=list()
  for(i in 1:length(est_clusters)){
    #[[1]]:loading, [[2]]:factor
    EST_CLUSTERS[[i]]=c(est_clusters[[i]][[1]][1]:est_clusters[[i]][[1]][2])
```

```
}
#---------------------
#  Recovery Score
#---------------------
sum=0
for(i in 1:length(TRUE_CLUSTERS)){
  max=0
  for(j in 1:length(EST_CLUSTERS)){
    and=length(intersect(TRUE_CLUSTERS[[i]], EST_CLUSTERS[[j]]))
    or=length(union(TRUE_CLUSTERS[[i]], EST_CLUSTERS[[j]]))
    rst=and/or
    if(rst > max)
      max=rst
  }
  sum=sum+max
}
rec=sum/length(TRUE_CLUSTERS)
#--------------------
#  Relevance Score
#--------------------
sum=0
for(i in 1:length(EST_CLUSTERS)){
  max=0
  for(j in 1:length(TRUE_CLUSTERS)){
    and=length(intersect(TRUE_CLUSTERS[[j]], EST_CLUSTERS[[i]]))
    or=length(union(TRUE_CLUSTERS[[j]], EST_CLUSTERS[[i]]))
    rst=and/or
    if(rst > max)
      max=rst
  }
  sum=sum+max
}
```

```
 rel=sum/length(EST_CLUSTERS)
 return (list(rec, rel))
}
Multiple_Runs_RR_Scores=function(OUTPUT_DIRS){

 #--------------------------------#
 #    Estimated Z and LAM
 #--------------------------------#
 #list sub directories and each sub contains outputs of one run of BicMix
 dirs=list.dirs(OUTPUT_DIRS, recursive=FALSE)
 n.dirs = length(dirs)

 RR_SCORES=NULL
 for (i in 1:n.dirs){
  z.est = as.matrix(read.table(paste(dirs[i],"/Z",sep=""), header = F))
  z.est=z.est[1,]
  o.est = as.matrix(read.table(paste(dirs[i],"/O",sep=""), header = F))
  o.est=o.est[1,]
  lam.est=as.matrix(read.table(paste(dirs[i],"/LAM",sep=""), header = F))
  #lam.est.sparse=lam.est[ ,z.est==1]
  x.est=as.matrix(read.table(paste(dirs[i],"/EX",sep=""), header = F))
  #x.est.sparse=x.est[o.est==1, ]

  if (length(which(z.est==1))==0 | length(which(o.est==1))==0){
   print("No sparse loading or factor has been discovered")
   next
  }
  est_clusters=get_clusters(lam.est, x.est, z.est, o.est)
  if (length(est_clusters)==0){
   next
 }
  scores=One_Run_RR_Score(true_clusters,est_clusters)
```

```
   RR_SCORES=rbind(RR_SCORES,unlist(scores))
  }
 return (RR_SCORES)
}
#-----------------------------------------------------
FILES_DIR=" C:/Users/Nick/Y/"
Y.TXT =paste(FILES_DIR,"Y.txt",sep="")
Z.TXT =paste(FILES_DIR,"Z.txt", sep ="")
O.TXT =paste(FILES_DIR,"O.txt",sep="")
LAM.TXT =paste(FILES_DIR,"LAM.txt",sep="")
X.TXT =paste(FILES_DIR,"X.txt",sep="")

#
#true biclusters
#
z.txt=as.matrix(read.table(Z.TXT, header = F))
z.txt=z.txt[1,]
o.txt=as.matrix(read.table(O.TXT, header = F))
o.txt=o.txt[1,]
lam.txt=as.matrix(read.table(LAM.TXT, header = F))
x.txt=as.matrix(read.table(X.TXT, header=F))
y.txt=as.matrix(read.table(Y.TXT, header=F))

#plot simulated Y
plot(density(y.txt), main="Simulated Data \n  Noise~N(0, 1)", xlab="Y")
plot(density(lam.txt), main="Simulated Data \n  Noise~N(0, 1)", xlab=expression(Lambda))
plot(density(x.txt), main="Simulated Data \n  Noise~N(0, 1)", xlab="X")
true_clusters=get_clusters(lam.txt, x.txt, z.txt, o.txt)
OUTPUT_DIRS="C:/Users/Nick/Desktop/2018 Red River Conference/Y/REF/"
SCORES = Multiple_Runs_RR_Scores(OUTPUT_DIRS)
Rec_avg=mean(SCORES[,1])
Rel_avg=mean(SCORES[,2])
```

```
#plot R&R scores
hist(SCORES[,1], main ="Distribution of R&R Scores",
    xlab ="Recovery Scores" )
hist(SCORES[,2], main ="Distribution of R&R Scores",
    xlab ="Relevance  Scores" )
plot(SCORES[,1], SCORES[,2], main="sim1-low noise", xlab="Recovery", ylab="Relevance",
    asp=0, xlim=c(0,1), ylim=c(0,1), col="red", lwd=2)
```

APPENDIX D. R CODE FOR DISTRIBUTION OF NUMBER OF GENES AND SAMPLES

```
OUTPUT_REF_DIRS="C:/BicMix_NKI_Output/results"
OUTPUT_DIRS = OUTPUT_REF_DIRS
dirs=list.dirs(OUTPUT_DIRS, recursive=FALSE)
n.dirs = length(dirs)
runs = n.dirs
#----------------------------------------
z_files= double(runs)
o_files = double(runs)
lam_files = double(runs)
ex_files = double(runs)
for(i in 1:runs){
  z_files[i] = paste("./result", i, "/Z", sep="")
  o_files[i] = paste("./result", i, "/O", sep="")
  lam_files[i] = paste("./result", i, "/LAM", sep="")
  ex_files[i]=paste("./result", i, "/EX", sep="")
}
z_diverge = 0
o_diverge = 0
# Z = double(runs)
# O = double(runs)
Z.data<-list()
O.data<-list()
z_index = 1
o_index = 1

z_components = 0
z_sparse = 0
o_components = 0
o_sparse = 0
```

```
for (i in 1:runs){
  # Z loadings
  if (file.exists(z_files[i])){
    Z.data[[z_index]] = read.table(z_files[i], header = F)
    compos = ncol(Z.data[[z_index]])
    sparse = length(which(Z.data[[z_index]][1,] == 1))
    z_components = z_components + compos
    z_sparse = z_sparse + sparse
    if (compos != sparse){
      print(paste("dense loadings recovered in file", z_files[i]))
    }
    z_index = z_index+1
  }
  else{
    z_diverge = z_diverge +1
  }
  #O factors
  if (file.exists(o_files[i])){
    O.data[[o_index]] = read.table(o_files[i], header = F)
    compos = ncol(O.data[[o_index]])
    sparse = length(which(O.data[[o_index]][1,] == 1))
    o_components = o_components + compos
    o_sparse = o_sparse + sparse
    if (compos != sparse)
      print(paste("dense factors recovered in file ", o_files[i]))

    o_index = o_index+1
  }else{
    o_diverge = o_diverge +1
  }
}
total_components = z_components + o_components
```

```
total_diverge = z_diverge + o_diverge
total_dense_loadings = z_components - z_sparse
total_dense_factors = o_components - o_sparse
#-------------------------------------------------------------------------------
# Number of Genes in Estimated Lambda Matrix
#-------------------------------------------------------------------------------

LAM.data<-list()
lam_index = 1
for (i in 1:runs){
  #LAM
  if (file.exists(lam_files[i])){
    LAM.data[[lam_index]] = read.table(lam_files[i], header = F)
    lam_index = lam_index + 1
  }
}

length(LAM.data)   # number of objects in the list
lengths(LAM.data)  # the length of each element in the list
index = 1
#lam_genes=double(sum(lengths(LAM.data)))
lam_genes=double(sum(sapply(LAM.data, NCOL)))

for (i in 1:length(LAM.data)){
  for (j in 1:ncol(LAM.data[[i]])){
    lam_genes[index] = length(which(LAM.data[[i]][,j] != 0))
    index = index + 1
  }
}
hist(sqrt(lam_genes), main ="Distribution of The Number of Genes",
     xlab ="sqrt(Genes in Loadings)" )
plot(density(sqrt(lam_genes)), main="Distribution of The Number of Genes",
```

```
    xlab = "sqrt(Genes in Loadings)")

#-----------------------------------------------------------------
# Number of Samples in Estimated Factor Matrix
#-----------------------------------------------------------------
EX.data<-list()
ex_index = 1
for (i in 1:runs){
  #EX
  if (file.exists(ex_files[i])){
    EX.data[[ex_index]] = read.table(ex_files[i], header = F)
    ex_index = ex_index + 1
  }
}
length(EX.data)   # number of objects in the list
index = 1
ex_samples=double(sum(sapply(EX.data, NROW)))
for (i in 1:length(EX.data)){
  for (j in 1:nrow(EX.data[[i]])){
    ex_samples[index] = length(which(EX.data[[i]][j, ] != 0))
    index = index + 1
  }
}
hist(sqrt(ex_samples), main="Distribution of The Number of Samples",
     xlab = "sqrt(Samples in Factors)")
plot(density(sqrt(ex_samples)), main="Distribution of The Number of Samples",
     xlab = "sqrt(Samples in Factors)")
```