Relaxing the Identically Distributed Assumption in Gaussian Co-Clustering for High Dimensional Data

M P B Gallaugher, C Biernacki, P D Mcnicholas

▶ **To cite this version:**

M P B Gallaugher, C Biernacki, P D Mcnicholas. Relaxing the Identically Distributed Assumption in Gaussian Co-Clustering for High Dimensional Data. 2018. hal-01862824v1

HAL Id: hal-01862824

https://hal.science/hal-01862824v1

Preprint submitted on 27 Aug 2018 (v1), last revised 21 Nov 2022 (v4)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Relaxing the Identically Distributed Assumption in Gaussian Co-Clustering for High Dimensional Data

M.P.B. Gallaugher, C. Biernacki, P.D. McNicholas

August 27, 2018

Abstract

A co-clustering model for continuous data that relaxes the identically distributed assumption within blocks of traditional co-clustering is presented. The proposed model, although allowing more flexibility, still maintains the very high degree of parsimony achieved by traditional co-clustering. A stochastic EM algorithm along with a Gibbs sampler is used for parameter estimation and an ICL criterion is used for model selection. Simulated and real datasets are used for illustration and comparison with traditional co-clustering.

1 Introduction

Clustering is the process of finding and analyzing underlying group structures in heterogenous data. With the emergence of big data, the number of variables in a dataset is constantly increasing and in many areas of application it is not uncommon for the number of variables to exceed the number of observations. In such situations where the dimension of the data is very high, traditional mixture modelling techniques for clustering oftentimes fail.

Co-clustering has become a very useful method for dealing with such scenarios. Co-clustering aims to define a partition in the rows of the data matrix for clustering individuals, as well as a partition in the columns for clustering variables. The result is partitioning the data matrix into homogenous blocks, or co-clusters, based on both individuals and variables. A key assumption for maintaining parsimony is that observations within each block are independent and identically distributed. Some of the earliest work in co-clustering was done by Hartigan (1972). Since that time, model-based approaches have recently been shown to be effective for continuous data, Nadif and Govaert (2010), count data, Pledger and Arnold (2014) and ordinal data, Jacques and Biernacki (2017), to only name a few.

In traditional co-clustering, added flexibility is obtained by fitting more clusters in rows and columns; however, this is not generally advisable for parsimonious reasons. Herein, we propose a co-clustering model for continuous data that separately clusters columns according to both means and variances using a Gaussian distribution. This effectively breaks the identically distributed assumption of observations within each block while still maintaining the parsimony of traditional co-clustering that makes it attractive for really high dimensional data.

The remainder of this paper is laid out as follows. Section 2 presents a detailed background on high dimensional clustering techniques as well as details on traditional co-clustering using the Gaussian distribution. Section 3 presents the new model, parameter estimation, model selection criterion, and a non-exhaustive search algorithm for model selection. In Sections 4 and 5, we look at the performance of the algorithms and parameter estimation as well as comparing the proposed model with traditional co-clustering on some synthetic datasets and on a real dataset respectively. We end with a discussion of the results (Section 6).

2 Gaussian-Based Clustering for High Dimensional Data

2.1 Model-Based Clustering

Clustering is the process of finding underlying group structures in a dataset $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)$ with n individuals $\mathbf{x}_i \in \mathbb{R}^p$. One common method for clustering is model-based clustering and this generally makes use of a finite mixture model. A finite mixture model assumes that a real random vector $\mathbf{X}_i$ of dimension p has probability density function

$$f(\mathbf{x}_i|\boldsymbol{\vartheta}) = \sum_{g=1}^{G} \pi_g f(\mathbf{x}_i|\boldsymbol{\Theta}_g),$$

where $\pi_g > 0, \forall g$ and $\sum_{g=1}^{G} \pi_g = 1$ are the mixing proportions and $f(\cdot|\boldsymbol{\Theta}_g)$ are the component density functions (among G) parameterized by $\boldsymbol{\Theta}_g$. $\boldsymbol{\vartheta}$ represents all the mixture parameters and is given by $\boldsymbol{\vartheta} = (\pi_1, \ldots, \pi_G, \boldsymbol{\Theta}_1, \ldots, \boldsymbol{\Theta}_G)$.

Because of its mathematical tractability, the multivariate Gaussian mixture model has been widely studied in the literature. In this case each of the component densities is a multivariate Gaussian with density

$$f(\mathbf{x}_i|\boldsymbol{\Theta}_g) = \frac{1}{(2\pi)^{\frac{p}{2}}|\boldsymbol{\Sigma}_g|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_g)'\boldsymbol{\Sigma}_g^{-1}(\mathbf{x}_i - \boldsymbol{\mu}_g)\right\},$$

where $\boldsymbol{\Theta}_g = (\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$. The number of parameters in a Gaussian mixture model is given by

$$\#\text{Params}_{\text{GaussMix}} = (G-1) + Gp + Gp(p+1)/2. \tag{1}$$

As is clearly evident, the number of parameters in this case is quadratic in the dimension of the data. As a result, using this simple mixture of Gaussian distributions will usually fail when the dimension, p, of the data reaches around 10.

In traditional model-based clustering, the group membership for observation $\mathbf{x}_i$ is usually represented by the vector $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iG})$, where $z_{ig} = 1$ if observation $\mathbf{x}_i$ belongs to group g and 0 otherwise. Moreover, $\mathbf{z}_i \sim \text{multinomial}(1; \boldsymbol{\pi})$ where $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_G)$. In addition, all couples $(\mathbf{X}_i, \mathbf{z}_i)$ are usually assumed to be independently drawed.

The earliest use of a Gaussian mixture model for clustering can be found in Wolfe (1965). Other early work on Gaussian mixture models for clustering can be found in Baum et al. (1970) and Scott and Symons (1971). A detailed review of model-based clustering and classification is given by McNicholas (2016), including related estimation and model selection procedures.

2.2 High Dimensional Clustering Techniques

Although the Gaussian mixture model is widely used, problems arise when moving to high dimensional datasets. The main impact of dimensionality is the number of parameters present in the component covariance matrices $\boldsymbol{\Sigma}_g$ which is quadratic in the dimension, see (1). Therefore many methods try to first impose parsimonious constraints on $\boldsymbol{\Sigma}_g$. A detailed background on this can be found in Bouveyron and Brunet-Saumard (2014).

One particular example to note is the mixture of factor analyzers model. This model was first presented by Ghahramani and Hinton (1997) and is a Gaussian mixture model with covariance structure $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}$, where $\boldsymbol{\Lambda}_g$ is a $p \times q$ matrix of factors with $q < p$. A small extension was presented by McLachlan and Peel (2000), who utilize the more general structure $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \boldsymbol{\Psi}_g$. Tipping and Bishop (1999) introduce the closely-related mixture of PPCAs with $\boldsymbol{\Sigma}_g = \boldsymbol{\Lambda}_g\boldsymbol{\Lambda}_g' + \psi_g\mathbf{I}$. McNicholas and Murphy (2008) constructed a family of eight parsimonious Gaussian models by considering the constraint $\boldsymbol{\Lambda}_g = \boldsymbol{\Lambda}$ in addition to $\boldsymbol{\Psi}_g = \boldsymbol{\Psi}$ and $\boldsymbol{\Psi}_g = \psi_g\mathbf{I}$. The main drawback to all these methods, is that although the number of parameters is reduced from quadratic to linear complexity in the dimension, it is still dependent on the dimension. For example, for the fully constrained model in McNicholas and Murphy (2008) the number of parameters is

$$\#\text{Param}_{\text{MFA}} = (G-1) + Gp + pq - q(q-1)/2 + 1. \tag{2}$$

Therefore, these models will not perform well on very high dimensional datasets. These methods also cannot be performed on datasets where $N > d$ which is common in applications such as gene expression data, word processing data, single nucleotide polymorphism (SNP) data, etc.

Alternatively, Bouveyron et al. (2007) consider using the spectral decomposition of $\boldsymbol{\Sigma}_g$

$$\boldsymbol{\Sigma}_g = \mathbf{D}_g \boldsymbol{\Delta}_g \mathbf{D}_g'$$

where $\mathbf{D}_g$ is the orthogonal matrix of eigenvectors and $\boldsymbol{\Delta}_g$ is a diagonal matrix of corresponding eigenvalues for which they impose the structure $\boldsymbol{\Delta}_g = \text{diag}(a_{1g}, a_{2g}, \ldots, a_{q_g g}, b_g, b_g, \ldots, b_g)$, where a_{kg} are the q_g largest eigenvalues and b_g is average of the remaining $p - q_g$ eigenvalues. This also greatly reduces the number of parameters, with the number of parameters given by

$$\#Param_{\text{Bouveyron}} = (G-1) + Gp + \sum_{g=1}^{G} q_g[p - (q_g+1)/2] + \sum_{g=1}^{G} q_g + 2G \tag{3}$$

Finally, there are also variable selection procedures such as ℓ_1 penalization methods which take advantage of sparsity to perform variable selection and parameter estimation simultaneously. The first such proposed method was presented by Pan and Shen (2007) which considered equal, diagonal covariance matrices between groups and applied an ℓ_1 penalty to the mean vectors. A lasso method is then used for parameter estimation. This was extended by Zhou et al. (2009) who considered unconstrained covariance matrices and applied an ℓ_1 penalty for both the mean and covariance parameters. Although these methods are useful for dealing with the dimensionality problem, as discussed by Meynet and Maugis-Rabusseau (2012), the ℓ_1 penalty shrinks the parameters, thus introducing bias. Moreover, the Bayesian information criterion (BIC; Schwarz, 1978) may not be suitable for high dimensional data. A detailed review of each of these methods can be found in Biernacki and Maugis (2017).

2.3 Co-Clustering and its Limits

Co-Clustering has become a very useful tool for high dimensional data. This method essentially considers simultaneous clustering of rows and columns and then organizes the data into blocks. As in clustering, in traditional co-clustering data is assumed to come in the form of an $n \times p$ matrix $\mathbf{x}$ with columns represented by $\mathbf{x}_i$. Each individual element of $\mathbf{x}_i$ will be denoted by x_{ij} so that x_{ij} is the observation in row i and column j.

Like in clustering, in co-clustering there is an unknown partition of the rows into G clusters that can be represented by the indicator vector $\mathbf{z}_i$ as defined previously. We also, symmetrically and unlike clustering, have a partition of the columns into L clusters that can

be represented by the indicator vector $\mathbf{w}_j = (w_{j1}, w_{j2}, \ldots, w_{jL}) \sim \text{multinomial}(1; \boldsymbol{\rho})$ where $w_{jl} = 1$ if column j belongs to column cluster l and 0 otherwise and $\boldsymbol{\rho} = (\rho_1, \rho_2, \ldots, \rho_L)$. It is assumed that each data point x_{ij} is independent once the $\mathbf{z}_i$ and $\mathbf{w}_j$ are fixed. If in addition, all $\mathbf{z}_i$ and $\mathbf{w}_j$ are independent, then utilizing the latent block model for continuous data and using the Gaussian distribution, as discussed in Nadif and Govaert (2010), for continuous data the density of $\mathbf{x}$ becomes $f(\mathbf{x}; \boldsymbol{\vartheta}) = \sum_{\mathbf{z} \in \mathcal{Z}} \sum_{\mathbf{w} \in \mathcal{W}} p(\mathbf{z}; \boldsymbol{\pi}) p(\mathbf{w}; \boldsymbol{\rho}) f(\mathbf{x} | \mathbf{z}, \mathbf{w}; \boldsymbol{\Theta})$ where $p(\mathbf{z}; \boldsymbol{\pi}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \pi_g^{z_{ig}}$, $p(\mathbf{w}; \boldsymbol{\rho}) = \prod_{j=1}^{p} \prod_{l=1}^{L} {\rho_l}^{w_{jl}}$, and

$$f(\mathbf{x} | \mathbf{z}, \mathbf{w}^{\mu}, \mathbf{w}^{\Sigma}; \boldsymbol{\Theta}) = \prod_{i=1}^{N} \prod_{g=1}^{G} \prod_{j=1}^{d} \prod_{l=1}^{L} \left[\frac{1}{\sqrt{2\pi}\sigma_{gl}} \exp \left\{ -\frac{1}{2\sigma_{gl}^2} (x_{ij} - \mu_{gl})^2 \right\} \right]^{z_{ig} w_{jl}},$$

where μ_{gl} and σ_{gl} are the mean and standard deviation respectively for row cluster g and column cluster l and $\boldsymbol{\Theta}$ is the set of all μ_{gl} and σ_{gl} and $\boldsymbol{\vartheta} = (\boldsymbol{\pi}, \boldsymbol{\rho}, \boldsymbol{\Theta})$ The total number of free parameters in this co-clustering model is $G + L + 2(GL - 1)$, which is not dependent on the dimension, making it a very parsimonious model with only the latent variables increasing with dimension. Moreover, co-clustering will still work when $N > d$.

Note that are two different ways that one can view co-clustering. The first is that the main goal is the clustering of rows, and the clustering of columns is solely a way to solve the problem of dimensionality. However, in certain applications, the clustering of the columns might also be of interest.

Limitations of Co-Clustering Although co-clustering has advantages over other high dimensional techniques (especially in the number of parameters), the model is fairly restrictive since each observation in a block is independent and identically distributed (iid) according to a Gaussian distribution with mean μ_{gl} and variance σ_{gl}^2. More flexibility is obtained by fitting more clusters in columns and lines, which is not always possible or advisable. What we propose in the present work is to relax the identically distributed assumption by clustering columns according to both mean and variance. This is the reason why we adopt hereafter the denomination "parameter-wise", that we present now in detail.

3 Parameter-Wise Gaussian Co-Clustering

3.1 A Model to Combine Two Latent Variables in Columns

Recall that traditional co-clustering aims to cluster data such that observations in the same block have the same distribution. An extension of traditional co-clustering for Gaussian data is now considered. Like in traditional co-clustering there is a partition in rows and columns,

but now there are two partitions in the columns, specifically a partition with respect to means and a partition with respect to variances.

Recall also that the continuous data is represented as an $n \times p$ matrix, $\mathbf{x} = (x_{ij})_{1 \leq i \leq n, 1 \leq j \leq p}$. The partition in rows is again represented by $\mathbf{z} = (\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n)$ where $\mathbf{z}_i$ is distributed the same as before.

Two Partitions in Columns The partition in columns by means is represented by $\mathbf{w}^{\mu} = (\mathbf{w}_1^{\mu}, \mathbf{w}_2^{\mu}, \ldots, \mathbf{w}_p^{\mu})$ where

$$\mathbf{w}_j^{\mu} = (w_{j1}^{\mu}, w_{j2}^{\mu}, \ldots, w_{jL^{\mu}}^{\mu}) \sim \text{multinomial}(1; \boldsymbol{\rho}^{\mu})$$

with $\boldsymbol{\rho}^{\mu} = (\rho_1^{\mu}, \rho_2^{\mu}, \ldots, \rho_{L^{\mu}}^{\mu})$ and also the partition in columns by variances is denoted by $\mathbf{w}^{\Sigma} = (\mathbf{w}_1^{\Sigma}, \mathbf{w}_2^{\Sigma}, \ldots, \mathbf{w}_p^{\Sigma})$ where

$$\mathbf{w}_j^{\Sigma} = (w_{j1}^{\Sigma}, w_{j2}^{\Sigma}, \ldots, w_{jL^{\Sigma}}^{\Sigma}) \sim \text{multinomial}(1; \boldsymbol{\rho}^{\Sigma}),$$

with $\boldsymbol{\rho}^{\Sigma} = (\rho_1^{\Sigma}, \rho_2^{\Sigma}, \ldots, \rho_{L^{\Sigma}}^{\Sigma})$ These two partitions in the columns is where the main novelty lies. Note that G, L^{μ} and L^{Σ} are the number of clusters in rows, columns by means, and columns by variances respectively.

Log-Likelihood Using a small extension of the latent block model the observed log-likelihood is then

$$f(\mathbf{x}; \boldsymbol{\vartheta}) = \sum_{\mathbf{z} \in Z} \sum_{\mathbf{w}^{\mu} \in W^{\mu}} \sum_{\mathbf{w}^{\Sigma} \in W^{\Sigma}} p(\mathbf{z}; \boldsymbol{\pi}) p(\mathbf{w}^{\mu}; \boldsymbol{\rho}^{\mu}) p(\mathbf{w}^{\Sigma}; \boldsymbol{\rho}^{\Sigma}) f(\mathbf{x} | \mathbf{z}, \mathbf{w}^{\mu}, \mathbf{w}^{\Sigma}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where

$$p(\mathbf{z}; \boldsymbol{\pi}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \pi_g^{z_{ig}},$$

$$p(\mathbf{w}^{\mu}; \boldsymbol{\rho}^{\mu}) = \prod_{j=1}^{p} \prod_{l^{\mu}=1}^{L^{\mu}} (\rho_{l^{\mu}}^{\mu})^{w_{jl^{\mu}}^{\mu}},$$

$$p(\mathbf{w}^{\Sigma}; \boldsymbol{\rho}^{\Sigma}) = \prod_{j=1}^{p} \prod_{l^{\Sigma}=1}^{L^{\Sigma}} (\rho_{l^{\Sigma}}^{\Sigma})^{w_{jl^{\Sigma}}^{\Sigma}},$$

and

$$f(\mathbf{x} | \mathbf{z}, \mathbf{w}^{\mu}, \mathbf{w}^{\Sigma}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \prod_{i=1}^{n} \prod_{g=1}^{G} \prod_{j=1}^{p} \prod_{l^{\mu}=1}^{L^{\mu}} \prod_{l^{\Sigma}=1}^{L^{\Sigma}} \left[\frac{1}{\sqrt{2\pi}\sigma_{gl^{\Sigma}}} \exp\left\{ -\frac{1}{2\sigma_{gl^{\Sigma}}^2} (x_{ij} - \mu_{gl^{\mu}})^2 \right\} \right]^{z_{ig} w_{jl^{\mu}}^{\mu} w_{jl^{\Sigma}}^{\Sigma}}.$$

In terms of notation, $\boldsymbol{\mu} = (\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \ldots, \boldsymbol{\mu}_G)$ where $\boldsymbol{\mu}_g = (\mu_{g1}, \mu_{g2}, \ldots, \mu_{gL^\mu})$. Note that μ_{gl^μ} is the mean for row cluster g and column cluster by means l^μ. Likewise, $\boldsymbol{\Sigma} = (\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2, \ldots, \boldsymbol{\Sigma}_G)$ where $\boldsymbol{\Sigma}_g = (\sigma^2_{g1}, \sigma^2_{g2}, \ldots, \sigma^2_{gL^\Sigma})$ and $\sigma^2_{gl^\Sigma}$ is the variance for row cluster g and column cluster by variances l^Σ. Finally, the complete-data log-likelihood is

$$p(\mathbf{x}, \mathbf{z}, \mathbf{w}^\mu, \mathbf{w}^\Sigma; \boldsymbol{\vartheta}) = C + \sum_{i=1}^{N}\sum_{g=1}^{G} z_{ig} \log \pi_g + \sum_{j=1}^{d}\sum_{l^\mu=1}^{L^\mu} w^\mu_{jl^\mu} \log \rho^\mu_{l^\mu} + \sum_{j=1}^{d}\sum_{l^\Sigma=1}^{L^\Sigma} w^\Sigma_{jl^\Sigma} \log \rho^\Sigma_{l^\Sigma} - \frac{1}{2}\sum_{i=1}^{N}\sum_{g=1}^{G}\sum_{j=1}^{d}\sum_{l^\mu=1}^{L^\mu}\sum_{l^\Sigma=1}^{L^\Sigma} z_{ig} w^\mu_{jl^\mu} w^\Sigma_{jl^\Sigma} \left[\log \sigma^2_{gl^\Sigma} + \frac{(x_{ij} - \mu_{gl^\mu})^2}{\sigma^2_{gl^\Sigma}}\right]$$

where C is a constant with respect to the parameters and $\boldsymbol{\vartheta} = (\boldsymbol{\pi}, \boldsymbol{\rho}^\mu, \boldsymbol{\rho}^\Sigma, \boldsymbol{\mu}, \boldsymbol{\Sigma})$. From this point on, we will refer to this model as non identically distributed (non-id) co-clustering.

Number of Free Parameters In the non-id co-clustering model the number of free parameters is

$$\begin{aligned} \#Param_{\text{new coclust}} &= G - 1 + L^\mu - 1 + L^\Sigma - 1 + GL^\mu + GL^\Sigma \\ &= G + (L^\mu + L^\Sigma)(G + 1) - 3. \end{aligned}$$

Therefore, like in traditional co-clustering, the number of free parameters is not dependent on the dimensionality of the data.

3.2 Parameter Estimation Using the SEM Gibbs Algorithm

The SEM algorithm after initialization at iteration q proceeds as follows.

SE Step: Generate the row partition $\mathbf{z}^{(q+1)}$ according to

$$P(z_{ig} = 1 | \mathbf{x}, \mathbf{w}^{\mu(q)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}^{(q)}, \boldsymbol{\Sigma}^{(q)}, \boldsymbol{\pi}^{(q)}) = \frac{\pi^{(q)}_g f(\mathbf{x}_i | \mathbf{w}^{\mu(q)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}^{(q)}_g, \boldsymbol{\Sigma}^{(q)}_g)}{\sum_{g'}^{G} \pi^{(q)}_{g'} f(\mathbf{x}_i | \mathbf{w}^{\mu(q)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}^{(q)}_{g'}, \boldsymbol{\Sigma}^{(q)}_{g'})}$$

where

$$f(\mathbf{x}_i | \mathbf{w}^{\mu(q)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}^{(q)}_g, \boldsymbol{\Sigma}^{(q)}_g) = \prod_{j=1}^{d}\prod_{l^\mu=1}^{L^\mu}\prod_{l^\Sigma=1}^{L^\Sigma} \left[\frac{1}{\sqrt{2\pi}\sigma^{(q)}_{gl^\Sigma}} \exp\left\{-\frac{1}{2\sigma^{2(q)}_{gl^\Sigma}}(x_{ij} - \mu^{(q)}_{gl^\mu})^2\right\}\right]^{w^{\mu(q)}_{jl^\mu} w^{\Sigma(q)}_{jl^\Sigma}}.$$

Generate the column partition by means $\mathbf{w}^{\mu(q+1)}$ according to

$$P(w^\mu_{jl^\mu} = 1 | \mathbf{x}, \mathbf{z}^{(q+1)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}^{(q)}, \boldsymbol{\Sigma}^{(q)}, \boldsymbol{\rho}^{\mu(q)}) = \frac{\rho^{\mu(q)}_{l^\mu} f(\mathbf{x}_{\cdot j} | \mathbf{z}^{(q+1)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}^{(q)}_{l^\mu}, \boldsymbol{\Sigma}^{(q)})}{\sum_{l^{\mu\prime}}^{L^\mu} \rho^{\mu(q)}_{l^{\mu\prime}} f(\mathbf{x}_{\cdot j} | \mathbf{z}^{(q+1)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}^{(q)}_{l^{\mu\prime}}, \boldsymbol{\Sigma}^{(q)})}$$

where $\mathbf{x}_{\cdot j} = (x_{1j}, x_{2j}, \ldots, x_{Nj})$, $\boldsymbol{\mu}_{l^\mu}^{(q)} = (\mu_{1l^\mu}^{(q)}, \mu_{2l^\mu}^{(q)}, \ldots, \mu_{Gl^\mu}^{(q)})$ and

$$f(\mathbf{x}_{\cdot j}|\mathbf{z}^{(q+1)}, \mathbf{w}^{\Sigma(q)}; \boldsymbol{\mu}_{l^\mu}^{(q)}, \boldsymbol{\Sigma}^{(q)}) = \prod_{i=1}^{N}\prod_{g=1}^{G}\prod_{l^\Sigma=1}^{L^\Sigma}\left[\frac{1}{\sqrt{2\pi}\sigma_{gl^\Sigma}^{(q)}}\exp\left\{-\frac{1}{2\sigma^{2(q)}_{gl^\Sigma}}(x_{ij}-\mu_{gl^\mu}^{(q)})^2\right\}\right]^{z_{ig}^{(q+1)}w^{\Sigma(q)}_{jl^\Sigma}}.$$

Generate the column partition by variances $\mathbf{w}^{\Sigma(q+1)}$ according to

$$P(w_{jl^\Sigma}^{\Sigma}=1|\mathbf{x}, \mathbf{z}^{(q+1)}, \mathbf{w}^{\mu(q+1)}; \boldsymbol{\mu}^{(q)}, \boldsymbol{\Sigma}^{(q)}, \boldsymbol{\rho}^{\Sigma(q)}) = \frac{\rho_{l^\Sigma}^{\Sigma(q)} f(\mathbf{x}_{\cdot j}|\mathbf{z}^{(q+1)}, \mathbf{w}^{\mu(q+1)}; \boldsymbol{\mu}^{(q)}, \boldsymbol{\Sigma}_{l^\Sigma}^{(q)})}{\sum_{l^{\Sigma\prime}}^{L^\Sigma}\rho_{l^{\Sigma\prime}}^{\Sigma(q)} f(\mathbf{x}_{\cdot j}|\mathbf{z}^{(q+1)}, \mathbf{w}^{\mu(q+1)}; \boldsymbol{\mu}^{(q)}, \boldsymbol{\Sigma}_{l^{\Sigma\prime}}^{(q)})}$$

where $\boldsymbol{\Sigma}_{l^\Sigma}^{(q)} = (\sigma^{2(q)}_{1l^\Sigma}, \sigma^{2(q)}_{2l^\Sigma}, \ldots, \sigma^{2(q)}_{Gl^\Sigma})$ and

$$f(\mathbf{x}_{\cdot j}|\mathbf{z}^{(q+1)}, \mathbf{w}^{\mu(q+1)}; \boldsymbol{\mu}^{(q)}, \boldsymbol{\Sigma}_{l^\Sigma}^{(q)}) = \prod_{i=1}^{N}\prod_{g=1}^{G}\prod_{l^\mu=1}^{L^\mu}\left[\frac{1}{\sqrt{2\pi}\sigma_{gl^\Sigma}^{(q)}}\exp\left\{-\frac{1}{2\sigma^{2(q)}_{gl^\Sigma}}(x_{ij}-\mu_{gl^\mu}^{(q)})^2\right\}\right]^{z_{ig}^{(q+1)}w^{\mu(q+1)}_{jl^\mu}}.$$

M Step: Update the parameters according to

$$\pi_g^{(q+1)} = \frac{\sum_{i=1}^n z_{ig}^{(q+1)}}{n}, \qquad \rho_{l^\mu}^{\mu\ (q+1)} = \frac{\sum_{j=1}^p w_{jl^\mu}^{\mu\ (q+1)}}{p}, \qquad \rho_{l^\Sigma}^{\Sigma\ (q+1)} = \frac{\sum_{j=1}^p w_{jl^\Sigma}^{\Sigma\ (q+1)}}{p},$$

$$\mu_{gl^\mu}^{(q+1)} = \frac{\sum_{i=1}^n\sum_{j=1}^p\sum_{l^\Sigma=1}^{L^\Sigma} z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)} w_{jl^\Sigma}^{\Sigma\ (q+1)} x_{ij}}{\sum_{i=1}^n\sum_{j=1}^p\sum_{l^\Sigma=1}^{L^\Sigma} z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)} w_{jl^\Sigma}^{\Sigma\ (q+1)}}$$

$$= \frac{\sum_{i=1}^n\sum_{j=1}^p z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)} x_{ij}}{\sum_{i=1}^N\sum_{j=1}^p z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)}},$$

$$\sigma^{2\ (q+1)}_{gl^\Sigma} = \frac{\sum_{i=1}^n\sum_{j=1}^p\sum_{l^\mu=1}^{L^\mu} z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)} w_{jl^\Sigma}^{\Sigma\ (q+1)} (x_{ij}-\mu_{gl^\mu}^{(q+1)})^2}{\sum_{i=1}^n\sum_{j=1}^p\sum_{l^\mu=1}^{L^\mu} z_{ig}^{(q+1)} w_{jl^\mu}^{\mu\ (q+1)} w_{jl^\Sigma}^{\Sigma\ (q+1)}}.$$

After a burn in period of the algorithm, the estimates of each of the parameters is just the mean of the runs of the SEM algorithm (the number of runs will be assessed experimentally in Section 4). We denote these final estimates by $\hat{\boldsymbol{\vartheta}} = (\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\rho}^\mu}, \hat{\boldsymbol{\rho}^\Sigma}, \hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\Sigma}})$ For the final partition of rows, columns by means, and columns by variances, we fix the parameters at their estimates, and run more iterations of the SE step. The average of these partitions over these additional runs is calculated and the partition is then taken to be the maximum a posteriori estimates. For our simulations and real data analysis, we took 20 such runs to obtain the final partitions $\hat{\mathbf{z}}, \hat{\mathbf{w}^\mu}$, and $\hat{\mathbf{w}^\Sigma}$.

3.3 Model Selection

ICL-BIC In the clustering scenario, the number of clusters in rows and columns by both means and variances is not known and therefore a model selection criterion is required. Like in traditional co-clustering, the observed log-likelihood is intractable and therefore the Bayesian information criterion (BIC; Schwarz, 1978) cannot be used. Therefore, we propose using the integrated complete log-likelihood (ICL; Biernacki et al., 2000) which relies on the complete data log-likelihood instead of the observed log-likelihood. This criterion will be called ICL-BIC, similar to that used in Jacques and Biernacki (2017) and it is given by

$$\text{ICL–BIC} = p(\mathbf{x}, \hat{\mathbf{z}}, \hat{\mathbf{w}^{\mu}}, \hat{\mathbf{w}^{\Sigma}}; \hat{\boldsymbol{\vartheta}}) - \frac{G-1}{2}\log N - \frac{L^{\mu}+L^{\Sigma}-2}{2}\log d - \frac{G(L^{\mu}+L^{\Sigma})}{2}\log Nd.$$

From the property proven by Brault et al. (2017), the BIC and ICL-BIC exhibit the same behaviour for large values of n and/or p, thus the number of blocks chosen by this criterion is consistent (under some conditions not mentioned here).

Search Algorithm Because an extra layer of complexity, namely the two different partitions of the columns, is introduced with the non-id model, it may be take a very long time to perform an exhaustive search of all possible combinations of G, L^{μ} and L^{Σ} in a pre-defined range. This has been discussed in the literature, namely Robert (2017). Specifically, the start values are set to $(G, L^{\mu}, L^{\Sigma}) = (G_1, L_1^{\mu}, L_1^{\Sigma})$. We then fit three models with parameters $(G_1+1, L^{\mu}, L^{\Sigma})$, $(G_1, L^{\mu}+1, L^{\Sigma})$ and $(G_1, L^{\mu}, L^{\Sigma}+1)$. The set with the highest ICL is retained and we get the set $(G_2, L_2^{\mu}, L_2^{\Sigma})$. The procedure is then repeated until a maximum threshold is reached for these parameters or the ICL no longer increases.

4 Numerical Experiments on Artificial Data

4.1 Algorithm and Parameter Estimation Evaluation

To evaluate the algorithm, we performed two simulations with different degrees of separation between the blocks. The purpose of these simulations was to look at the number of iterations of the SEM algorithm that is needed for parameter estimation and classification performance.

Simulation 1

The first simulation had good separation between the blocks. We took $N = 1000$, $d = 100$, $G = 3$, $L^{\mu} = 2$, $L^{\Sigma} = 3$. The means and variances were respectively taken to be

$$\boldsymbol{\mu} = \begin{pmatrix} 1 & -1 \\ 2 & -2 \\ 3 & -3 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.75 \\ 2 & 1.75 & 0.25 \\ 1.5 & 2.25 & 2.5 \end{pmatrix}.$$

Note that each cell of the matrices corresponds to the parameter for the corresponding cluster in lines and in columns. So for example, the first row first column element of $\boldsymbol{\mu}$ represents the mean parameter for the first cluster in lines and the first cluster in columns according to the mean. The mixing proportions were taken to be

$$\boldsymbol{\pi} = (0.3, 0.3, 0.4), \qquad \boldsymbol{\rho}^{\mu} = (0.4, 0.6), \qquad \boldsymbol{\rho}^{\Sigma} = (0.3, 0.3, 0.4).$$

50 datasets were simulated according to these parameters. A burn-in of 20 iterations of the SEM-Gibbs algorithm was used, followed by 100 iterations, followed by 20 iterations of the SE step for the final partitions.

In Table 1, we display the average errors of the estimates, up to a label switch. We measure the mean estimate quality by $\Delta\boldsymbol{\mu} = \sum_{g,l^{\mu}} |\hat{\mu}_{gl^{\mu}} - \mu_{gl^{\mu}}|$ and similarly for the other parameters. The errors are pretty small indicating that the parameter estimates from the algorithm are generally close to the true values. Table 2 displays the average ARI for the row, column by means and column by variance partitions. Notice that the classification is perfect for both partitions by columns for all 50 datasets. Moreover, the average ARI for the rows is also very high.

Table 1: Average error (and standard deviation) of the estimates over the 50 datasets for Simulation 1.

$\Delta\boldsymbol{\mu}$	$\Delta\boldsymbol{\Sigma}$	$\Delta\boldsymbol{\pi}$	$\Delta\boldsymbol{\rho}^{\mu}$	$\Delta\boldsymbol{\rho}^{\Sigma}$
0.14 (0.70)	0.24 (0.75)	0.012 (0.082)	1.44e-15 (5.61e-16)	1.33e-15 (4.59e-16)

Table 2: Average ARI (and standard deviation) for the rows (ARI_r), column by means ($\text{ARI}_{c\mu}$), and column by variance ($\text{ARI}_{c\Sigma}$) partitions over the 50 datasets for Simulation 1.

ARI_r	$\text{ARI}_{c\mu}$	$\text{ARI}_{c\Sigma}$
0.99 (0.068)	1.00 (0.00)	1.00 (0.00)

In Figure 1, we show the progression of the parameter estimates over the course of the SEM-Gibbs algorithm for one of our datasets (all other datasets exhibit similar behaviour). Notice that after a burn in period of 20, we obtain a very stable chain.

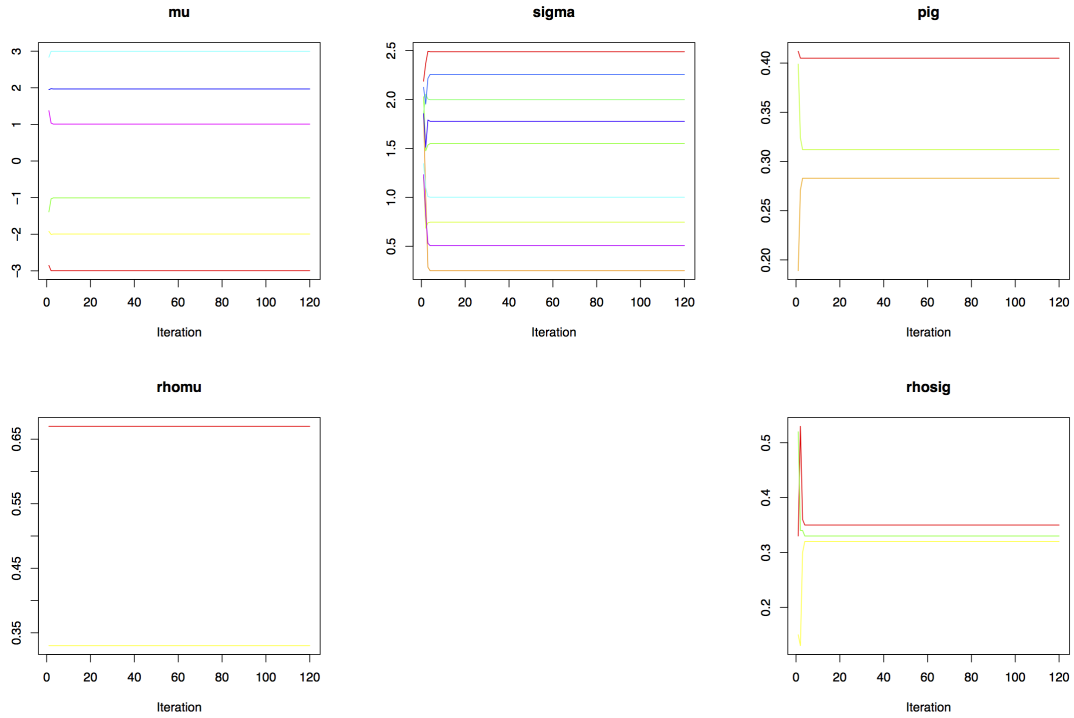

Figure 1: Simulation 1 SEM algorithm estimation progression.

Finally, in Figure 2, we display the true clustering result (which in this case is equivalent to the estimate clustering result) for one of the 50 datasets. In the co-clustering by means panel, the co-clustering results for the row partition and the column partition by means is shown. The co-clustering by variances shows the co-clustering results for the row partition and the column partition by variances. Finally, the combined co-clustering looks at all combinations of l^{μ} and l^{Σ} and then organizes the columns so that each column in the same cluster have the same mean and variance as in regular co-clustering. Specifically as seen here, the first cluster in columns would be those columns that were partitioned into cluster 1 for the means and cluster 1 for the variances, the second cluster would be those clustered into cluster 1 for the means and cluster 2 for the variances and so on. In a general scenario, this would correspond to a maximum of $L^{\mu}L^{\Sigma}$ clusters in columns thus allowing more flexibility but not greatly increasing the number of parameters. It is important to note, however, that there may be cases, as we will see with the real dataset, where no columns would be clustered into a particular pair of l^{μ} and l^{Σ} and thus the combined co-clustering results might have

fewer $L^{\mu}L^{\Sigma}$ clusters but never more. From this figure it is clear that the blocks are well separated.

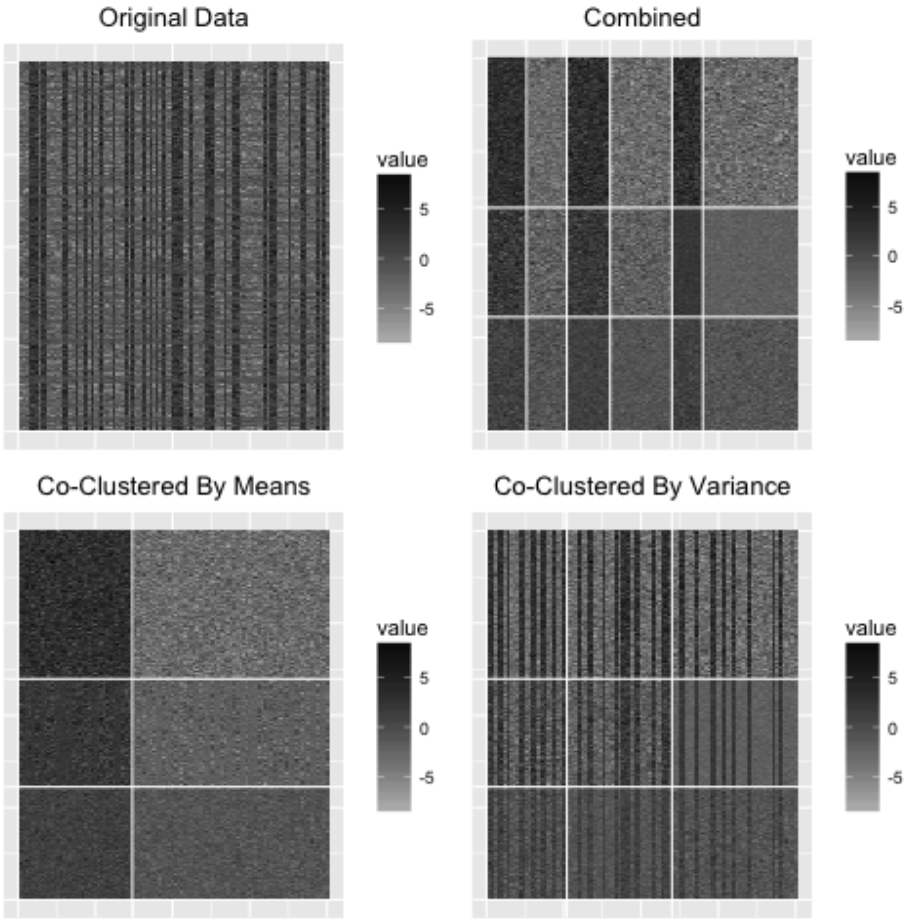

Figure 2: True co-clustering for one dataset from Simulation 1 which is the same as the predicted co-clustering results for this dataset.

Simulation 2

In Simulation 2, less separation between groups was considered. This time we considered $n = 200$, $p = 500$, $G = 3$, $L^{\mu} = 3$, $L^{\Sigma} = 2$ and took the means and variances to be

$$\boldsymbol{\mu} = \begin{pmatrix} 1 & 1.25 & 0 \\ 2 & 1.2 & 1 \\ 1.5 & 1.9 & 0.5 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 \\ 2 & 1.75 \\ 1.5 & 2.25 \end{pmatrix},$$

and the mixing proportions were

$$\boldsymbol{\pi} = (0.3, 0.3, 0.4), \qquad \boldsymbol{\rho}^{\mu} = (0.3, 0.5, 0.2), \qquad \boldsymbol{\rho}^{\Sigma} = (0.4, 0.6).$$

The algorithm was performed in the same way as before. In Table 3 we show the average error of the estimates with their standard deviations like before. The ARI results are also shown in Table 4. Figure 3 shows the progression of the estimates. We see this time that although it is not as flat as that seen in Simulation 1, we still obtain a fairly stable chain after a burn in of around 20 iterations of the SEM algorithm. Finally, we display the true co-clustered data like before in Figure 4. We see here that unlike in the first simulation, there is little separation between blocks.

Table 3: Average error (and standard deviation) of the estimates over the 50 datasets for Simulation 2.

$\Delta\boldsymbol{\mu}$	$\Delta\boldsymbol{\Sigma}$	$\Delta\boldsymbol{\pi}$	$\Delta\boldsymbol{\rho}^{\Sigma}$	$\Delta\boldsymbol{\rho}^{\Sigma}$
0.15 (0.50)	0.085 (0.046)	1.29e-15 (3.91e-16)	0.015 (0.088)	0.0079 (0.0054)

Table 4: Average ARI (and standard deviation) for the rows (ARI_r), column by means ($\text{ARI}_{c\mu}$), and column by variance ($\text{ARI}_{c\Sigma}$) partitions over the 50 datasets for Simulation 2.

ARI_r	$\text{ARI}_{c\mu}$	$\text{ARI}_{c\Sigma}$
1.00 (0.00)	0.98 (0.080)	0.96 (0.018)

4.2 ICL-BIC Selection Criterion

In this simulation, we were interested in the performance of the ICL-BIC criterion selecting the correct number of partitions when using an exhaustive search. Here we considered 2000 observations with $p = 500$. We also choose 3 groups in lines, columns by means and columns by variances. The parameters were

$$\boldsymbol{\mu} = \begin{pmatrix} 1 & 1.25 & 0 \\ 2 & 1.2 & 1 \\ 1.5 & 1.9 & 0.5 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.25 \\ 2 & 1.75 & 0.5 \\ 1.5 & 2.25 & 1 \end{pmatrix},$$

and

$$\boldsymbol{\pi} = (0.3, 0.3, 0.4), \qquad \boldsymbol{\rho}^{\mu} = (0.3, 0.4, 0.3), \qquad \boldsymbol{\rho}^{\Sigma} = (0.4, 0.3, 0.3).$$

An exhaustive search was performed considering each combination with $G, L^{\mu}, L^{\Sigma} \in \{2, 3, 4\}$. In Table 5, we display the number of times each number of partitions was chosen by the ICL-BIC. Notice that except for 1 dataset for clusters in lines, and 2 for clusters in columns for means and variances, the correct number of clusters was chosen for all 50 datasets.

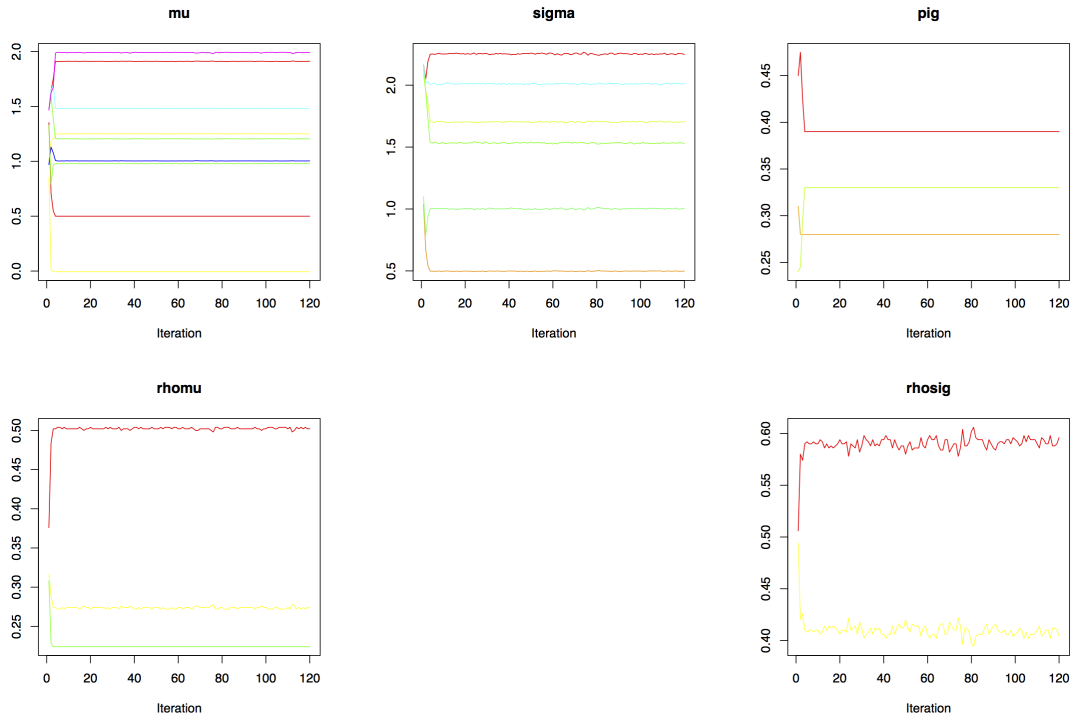

Figure 3: Simulation 2 SEM algorithm estimation progression.

Table 5: Frequency of the number of partitions chosen by the ICL-BIC over the 50 simulated datasets.

	2	3	4
G	0	49	1
L^{μ}	0	48	2
L^{Σ}	0	48	2

4.3 Search Algorithm Evaluation

The last simulation looked at the non-exhaustive search algorithm described in Section 3.3. There were 3 clusters in rows and columns by variances and 4 in columns by means. The parameters were taken to be

$$\boldsymbol{\mu} = \begin{pmatrix} 1 & -0.25 & 0.3 & -1 \\ 1.25 & 0 & 0.1 & -0.3 \\ 0.5 & -1 & 0 & 0.1 \end{pmatrix}, \qquad \boldsymbol{\Sigma} = \begin{pmatrix} 1 & 0.5 & 0.25 \\ 2 & 1.75 & 0.5 \\ 1.5 & 2.25 & 1 \end{pmatrix},$$

and

$$\boldsymbol{\pi} = (0.3, 0.3, 0.4), \qquad \boldsymbol{\rho}^{\mu} = (0.2, 0.3, 0.25, 0.25), \qquad \boldsymbol{\rho}^{\Sigma} = (0.5, 0.25, 0.25).$$

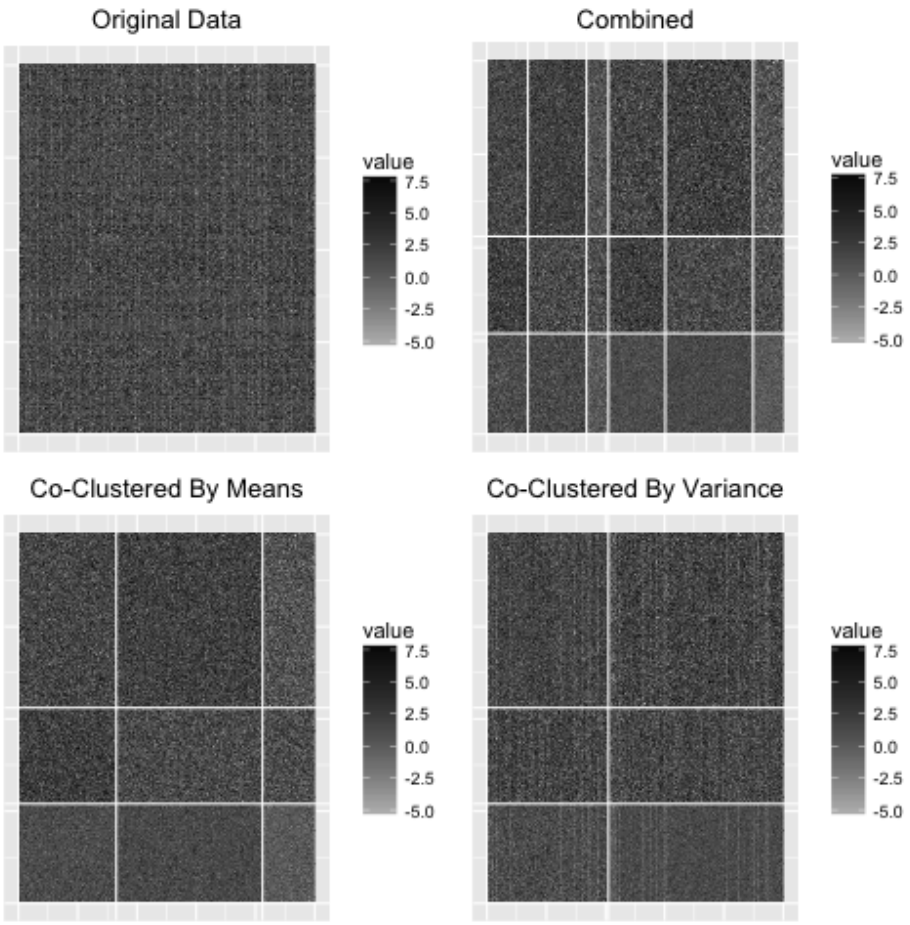

Figure 4: True co-clustering for one dataset from Simulation 2 which is the same as the predicted co-clustering results for this dataset.

The procedure was performed for 25 datasets and with initial values of $(G_1, L_1^{\mu}, L_1^{\Sigma}) = (1, 1, 1)$ and the maximum values for all three were set to 5. In Table 6 we show the number of times each group was chosen using this non-exhaustive procedure. We see that the procedure performs quite well with choosing the correct number of groups.

Table 6: Frequency of the number of clusters chosen by the ICL-BIC over the 25 simulated datasets when using the non-exhaustive search method.

	2	3	4
G	0	24	1
L^{μ}	0	25	0
L^{Σ}	1	24	0

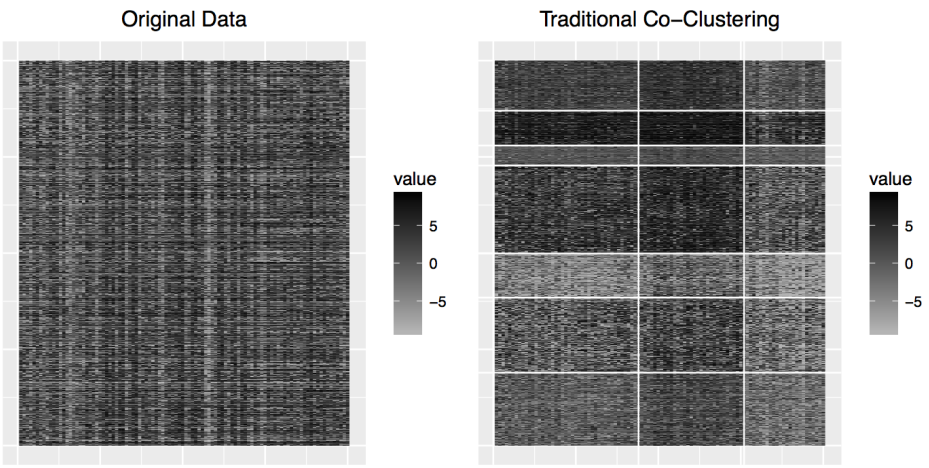

Figure 5: Traditional co-clustering results for the Jokes data.

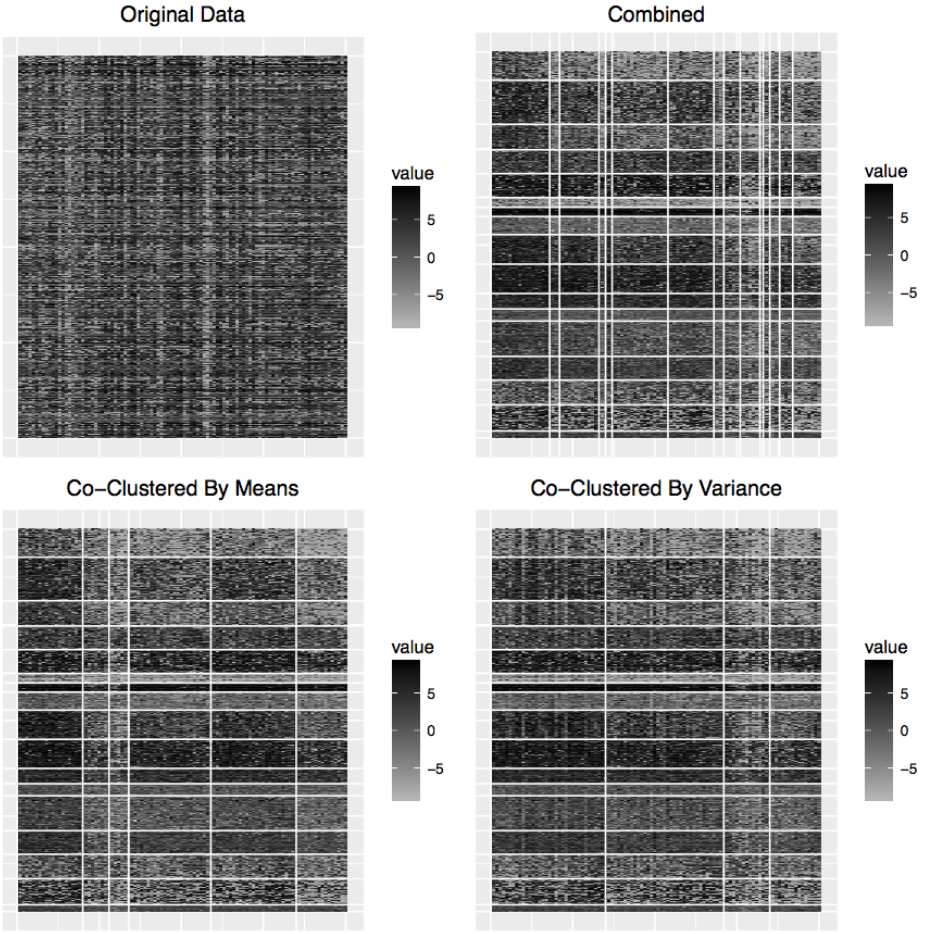

Figure 6: Non-Id co-clustering results for the Jokes data.

5 Real Data Analysis

We consider the Jester dataset used by Goldberg et al. (2001) and use this to compare the non-id co-clustering method with traditional co-clustering. Users gave 100 jokes a continuous

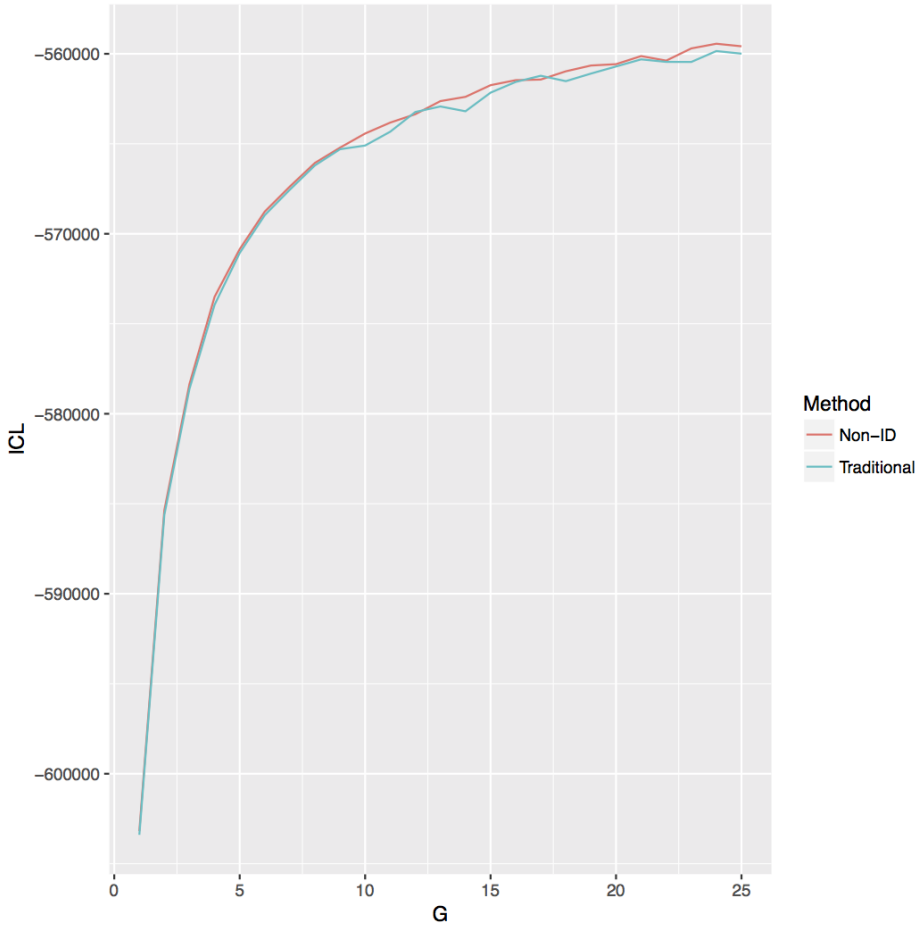

Figure 7: ICL-BIC results for traditional and non-id co-clustering when using the exhaustive search algorithm for the Jokes data.

rating from -10 to +10. A total of 7200 users rated all 100 jokes. We took a random sample of 2000 to make up our dataset. The non-id model was fitted for 1 to 25 groups in lines and 1 to 7 groups for means and variances in columns. Traditional co-clustering was performed for 1 to 25 groups in lines and 1 to 10 groups in columns.

When the non-exhaustive search algorithm was used the model selected using traditional co-clustering had 7 groups in rows and 3 groups in columns, leading to 50 estimated parameters. The resulting ICL-BIC was -569487. For non-id co-clustering, the final model had 17 groups in rows, 6 groups in columns by means and 4 groups in columns by variances with a total of 194 estimated parameters. The resulting ICL-BIC was -561099 and the total number of groups in the combined co-clustering was 15. In Figure 5 and 6 we show the visualization of the traditional co-clustering results and new method co-clustering results respectively. Due to the ICL-BIC choosing more groups in lines for the new method we propose, we can see that there is more heterogeneity in the users than was seen when using

traditional co-clustering. For example, the 6th and 7th groups in lines from the top appear to be individuals who consistently rated all jokes high or all jokes low with very little variation. Moreover, the group at the very bottom appears to be individuals who consistently ranked the jokes in the middle. Although the traditional co-clustering seems to separate individuals in a similar fashion (groups 2, 3 and 5 from the top), there is clearly more variability. It is also interesting to point out that for the non-id model a different number of groups in columns for means and variances were chosen again displaying the increased flexibility of the non-id model. Finally, this very large increase in flexibility is obtained without a drastic increase in the number of estimated parameters (a difference of 144) and still obtaining a higher ICL-BIC than traditional co-clustering.

It is important to note that this example also illustrates the necessity of considering the visualization of the means and the variances separately as well as the combined co-clustering results when using the non-id method. In the combined co-clustering visualization, it is a little difficult to see some of the clusters in columns because the number of columns in each cluster is very small; however, the two separate co-clusterings by means and variances in the second row provide a clearer visual. In this particular application the interpretation for these two additional visuals is a little difficult since there isn't much information about the jokes, in an application such as gene expression data,

Finally, in Figure 7, we display a plot of the highest ICL-BIC over all L for traditional co-clustering and L^{μ} and L^{Σ} for non-id co-clustering against G (the number of groups in rows) when using the exhaustive search algorithm. Although not displayed here, the values of L for traditional and L^{μ} and L^{Σ} for non-id co-clustering that resulted in the best ICL have a lot of variation. Moreover, for traditional co-clustering we can see from the graph that there is a fair amount of variation in the ICL-BIC once G approaches 10, and for non-iid co-clustering this occurs when G is approximately 15. Therefore, when using the exhaustive search algorithm, it can be difficult to choose the number of groups in lines and also in columns for both traditional and non-iid co-clustering. Moreover, it is very computationally expensive to run the exhaustive search with the non-iid co-clustering taking a little over 24 hours using 25 1200MHz cores running continuously.

6 Discussion

In this paper, we developed a co-clustering model that allowed more flexibility without greatly increasing the number of parameters by partitioning the columns of a continuous data matrix by both mean and variances. This in effect relaxes the identically distributed assumption of traditional co-clustering. Parameter estimation was carried out by the SEM-

Gibbs algorithm and an ICL criterion was used to choose the number of blocks. In addition, due to the increased number of possibilities when choosing the number of blocks, we proposed a non-exhaustive search method that was shown to perform well for both simulated and real data.

When analyzing the jokes dataset, the non-id method displayed three main advantages to traditional co-clustering. The first is the increase in flexibility while still maintaining parsimony. Specifically, we obtained a total of 15 groups in columns using the non-id method, compared to 3 when using traditional co-clustering resulting in an additional 144 estimated parameters. This difference in parameters was also based on 17 groups in rows for non-id co-clustering verses 7 for traditional co-clustering. It is worth noting that if G was kept constant at 7 between the two methods with the same partitions in columns, there would only be an additional 34 parameters when using non-id co-clustering. The non-id method also displayed easier interpretation. Because the number of groups in columns according to means and variances are less numerous than the total number of groups in columns in the combined co-clustering, and traditional co-clustering when increasing L, looking at the partitions according to means and variances offer a simplified visualization. This was particularly evident in the jokes dataset as the two separate partitions offered a simplified visualization when compared to the combined co-clustering with 15 groups in columns. Finally, a better ICL-BIC was achieved.

Although this method dealt with continuous data using the Gaussian distribution, it can be extended in various ways. One example would be to use different distributions with more than one parameter. For example, one could consider using distributions that account for tail weight, such as the t-distribution, or skewness and tail weight like the skew-t, variance gamma, or generalized hyperbolic distributions. In these cases, one could take a partition in the columns according to location, scale, concentration and skewness. This could also be applied to data that is not continuous like ordinal data where the columns could be partitioned according to mode and precision. Again, the number of parameters in each of these cases will not depend on the dimensionality of the data thus preserving the parsimony that is inherent to co-clustering.

References

Baum, L. E., Petrie, T., Soules, G. and Weiss, N. (1970), 'A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains', *Annals of Mathematical Statistics* **41**, 164–171.

Biernacki, C., Celeux, G. and Govaert, G. (2000), 'Assessing a mixture model for clustering with the integrated completed likelihood', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**(7), 719–725.

Biernacki, C. and Maugis, C. (2017), High-dimensional clustering, *in* 'Choix de modèles et agrégation, Sous la direction de J-J. DROESBEKE, G. SAPORTA, C. THOMAS-AGNAN Edition: Technip.'.
URL: *https://hal.archives-ouvertes.fr/hal-01252673*

Bouveyron, C. and Brunet-Saumard, C. (2014), 'Model-based clustering of high-dimensional data: A review', *Computational Statistics and Data Analysis* **71**, 52–78.

Bouveyron, C., Girard, S. and Schmid, C. (2007), 'High-dimensional data clustering', *Computational Statistics and Data Analysis* **52**(1), 502–519.

Brault, V., Keribin, C. and Mariadassou, M. (2017), 'Consistency and asymptotic normality of latent blocks model estimators', *arXiv preprint arXiv:1704.06629* .

Ghahramani, Z. and Hinton, G. E. (1997), The EM algorithm for factor analyzers, Technical Report CRG-TR-96-1, University of Toronto, Toronto, Canada.

Goldberg, K., Roeder, T., Gupta, D. and Perkins, C. (2001), 'Eigentaste: A constant time collaborative filtering algorithm', *information retrieval* **4**(2), 133–151.

Hartigan, J. A. (1972), 'Direct clustering of a data matrix', *Journal of the American statistical association* **67**(337), 123–129.

Jacques, J. and Biernacki, C. (2017), Model-Based Co-clustering for Ordinal Data. working paper or preprint.
URL: *https://hal.inria.fr/hal-01448299*

McLachlan, G. J. and Peel, D. (2000), Mixtures of factor analyzers, *in* 'Proceedings of the Seventh International Conference on Machine Learning', Morgan Kaufmann, San Francisco, pp. 599–606.

McNicholas, P. D. (2016), 'Model-based clustering', *Journal of Classification* **33**.

McNicholas, P. D. and Murphy, T. B. (2008), 'Parsimonious Gaussian mixture models', *Statistics and Computing* **18**(3), 285–296.

Meynet, C. and Maugis-Rabusseau, C. (2012), A sparse variable selection procedure in model-based clustering, Research report.
URL: *https://hal.inria.fr/hal-00734316*

Nadif, M. and Govaert, G. (2010), Model-based co-clustering for continuous data, *in* 'Machine Learning and Applications (ICMLA), 2010 Ninth International Conference on', IEEE, pp. 175–180.

Pan, W. and Shen, X. (2007), 'Penalized model-based clustering with application to variable selection', *Journal of Machine Learning Research* **8**(May), 1145–1164.

Pledger, S. and Arnold, R. (2014), 'Multivariate methods using mixtures: Correspondence analysis, scaling and pattern-detection', *Computational Statistics & Data Analysis* **71**, 241–261.

Robert, V. (2017), Coclustering for the analysis of pharmacovigilance large datasets, Theses, Université Paris Saclay ; Université Paris Sud - Orsay.
URL: *https://hal.inria.fr/tel-01695568*

Schwarz, G. (1978), 'Estimating the dimension of a model', *The Annals of Statistics* **6**(2), 461–464.

Scott, A. J. and Symons, M. J. (1971), 'Clustering methods based on likelihood ratio criteria', *Biometrics* **27**, 387–397.

Tipping, M. E. and Bishop, C. M. (1999), 'Mixtures of probabilistic principal component analysers', *Neural Computation* **11**(2), 443–482.

Wolfe, J. H. (1965), A computer program for the maximum likelihood analysis of types, Technical Bulletin 65-15, U.S. Naval Personnel Research Activity.

Zhou, H., Pan, W. and Shen, X. (2009), 'Penalized model-based clustering with unconstrained covariance matrices', *Electronic journal of statistics* **3**, 1473.