



HAL
open science

MMM: Clustering Multivariate Longitudinal Mixed-type Data

Francesco Amato, Julien Jacques

► **To cite this version:**

Francesco Amato, Julien Jacques. MMM: Clustering Multivariate Longitudinal Mixed-type Data. 2024. hal-04807626

HAL Id: hal-04807626

<https://hal.science/hal-04807626v1>

Preprint submitted on 27 Nov 2024

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.

MMM: CLUSTERING MULTIVARIATE LONGITUDINAL MIXED-TYPE DATA

Francesco Amato ¹, Julien Jacques ¹

¹ *Univ Lyon, Univ Lyon 2, ERIC, Lyon.*
{francesco.amato, julien.jacques}@univ-lyon2.fr

Abstract. Multivariate longitudinal data of mixed-type are increasingly collected in many science domains. However, algorithms to cluster this kind of data remain scarce, due to the challenge to simultaneously model the within- and between-time dependence structures for multivariate data of mixed kind. We introduce the Mixture of Mixed-Matrices (MMM) model: reorganizing the data in a three-way structure and assuming that the non-continuous variables are observations of underlying latent continuous variables, the model relies on a mixture of matrix-variate normal distributions to perform clustering in the latent dimension. The MMM model is thus able to handle continuous, ordinal, binary, nominal and count data and to concurrently model the heterogeneity, the association among the responses and the temporal dependence structure in a parsimonious way and without assuming conditional independence. The inference is carried out through an MCMC-EM algorithm, which is detailed. An evaluation of the model through synthetic data shows its inference abilities. A real-world application on financial data is presented.

Keywords. Model-based clustering. Mixed-type multivariate longitudinal data. Three-way data. Mixture models. Matrix-variate Gaussians.

1 Context

Multivariate longitudinal data of mixed-type are increasingly collected in many science domains. For example, in social sciences studies are often based on questionnaires encompassing different type of answers completed by participants multiple times. In physical sciences, phenomena are often measured repeatedly with different types of measurements. However, the statistical analysis of these data is far from simple, for several reasons. First, the collected data are often of different typology, ranging from continuous to count data. The analysis of such mixed-type data is a current research problem in the fields of statistics and machine learning ([Ahmad et al., 2019](#)). The second scientific obstacle is the modeling of the temporal trajectory. Currently, frequently the analyses are done independently at each temporal phase, then researchers try *a posteriori* to find links among times, by seeking from one phase to the other to find similar typical behavior. An example is [Selosse, Jacques, Biernacki, and Cousson-Glie, 2019](#) in the case of clustering of

longitudinal ordinal data for an application in psychology.

In this work we aim at providing a tool to perform clustering on multivariate longitudinal mixed-type data. Probabilistic (or model-based) clustering offers the advantage of clearly stating the assumptions behind the clustering algorithm, and allows cluster analysis to benefit from the inferential framework of statistics to address some of the practical questions arising when performing clustering (Bouveyron et al., 2019).

1.1 Related work

While several approaches exist for the clustering longitudinal and mixed-type data separately, literature is rather poor when they are to be dealt with simultaneously. In the following, we will present a brief overlook to the main methods to cluster mixed data, longitudinal data and mixed longitudinal data.

Although many data sets contain mixed-type data, few mixture models can manage these data (Hunt et al., 2011) due to the shortage of multivariate distributions able to handle them. Clustering with mixed-type data have received a large attention in the last decade from the researcher in statistics and machine learning. The latent class model (Everitt, 1984) is frequently used, and it assumes that the variables are conditionally independent upon the cluster membership. Consequently, the joint probability distribution function (pdf) of the features of different types is obtained by the product of the pdfs of each individual feature. However, when the variables are inherently correlated in a cluster, this model is not suitable. To overcome this issue, the authors of Marbac et al., 2017 wanted to conserve standard marginal distributions but also try to loosen the conditional independence on the variables. For this purpose, they use a copula, which allow definition of both the dependence model and the type of marginal distributions. The proposed model relies on the main assumption that each cluster follows a Gaussian copula. However, the authors note that model complexity increases promptly with the number of variables, which is not suitable in a big-data context. Moreover, it is not easily interpretable by non-statistician practitioners. More recently, Hermes et al., 2024 proposed a similar approach by using copulas in the context of graphical models, which were already extended for use for mixed-type data by Cheng et al., 2017. In Selosse, Jacques, and Biernacki, 2020, another model-based approach for ordinal, nominal, integer and continuous data is proposed, on the basis of conditional independence assumption and with the particularity of creating clusters of features as well as clusters of individuals (co-clustering).

Another way to address the issues of mixed-type data is to see some variables as the manifestation of latent variables. For example, in McParland et al., 2016, the clustMD model considers continuous and categorical data (nominal and ordinal) and assumes that a categorical variable is the representation of an underlying latent continuous variable.

Then, it is assumed that the continuous variables (observed and unobserved) follow a multivariate Gaussian mixture model. This model is further developed to address sparsity by [Choi et al., 2023](#).

Modeling longitudinal data poses a different kind of challenge than mixed-type data, as the grouping has to account for the similarity of individual trajectories which disrupt the independence assumption among observations. Additionally, this kind of data introduces the issue of dealing with time, often with sparse observations that makes unsuitable the use of models coming from the domains such as functional data, time series and Gaussian processes. In order to bypass these problems, some authors preferred to focus on geometric non-parametric clustering algorithms, as done by [Bruckers et al., 2016](#) with an idea based on k-means clustering and by [Zhou et al., 2023](#) with hierarchical clustering, among others. For parametric methods, a well established manner to model longitudinal data is through mixed-effects models. This research domain is well-established and vast. We refer to [Gad et al., 2012](#) for an overview and to the related work section of [Hui et al., 2024](#) for the most recent advancements. The main issues with this kind of models are the over-parametrization and the computational burden that often arises with it.

Another approach to clustering longitudinal data that gained traction in the last decade consists in arranging the data in a three-way format and modeling them through a matrix-variate mixture model. This approach offers the advantage of accounting for the overall time-behavior, grouping together the units that have a similar pattern across and within time. While not being new ([Basford et al., 1985](#)), matrix-variate distributions have recently gained attention, and mixtures of matrix-normals (MMN) have been developed and applied both in a frequentist framework in [Viroli, 2011a](#) and within a Bayesian one by [Viroli, 2011b](#). These models represent a natural extension of the multivariate normal mixtures to account for temporal (or even spatial) dependencies, and have the advantage of being also relatively easy to estimate by means of EM algorithm (a nice short description of the EM application to MNN is provided in §2.1 of [Wang et al., 2020](#)). In addition, in the context of linear mixed models with discrete individual random intercepts to analyze longitudinal continuous data, [Anderlucci et al., 2015](#) proposed Covariance Pattern Mixture Model (CPMM) which, by leveraging three-way data structures, does not require the usual local independence assumption. This model can be seen as an extension of the proposal of [McNicholas et al., 2010](#) in the multivariate context. More recently, in [Gallaughner et al., 2018](#) and [Melnykov et al., 2018, 2019](#) extensions for non-normal skewed cases have been proposed and applied. However, matrix-variate models suffer from over-parametrization that leads to estimation issues. To overcome this issue a more parsimonious model ([Sarkar et al., 2020](#)) and a new R package ([Zhu et al., 2022](#)) has been proposed. In addition, [Cappozzo et al., 2023](#) proposed a lasso-type penalization to account for sparsity. Despite their efficacy, up to now these methods have generally only been applied to continuous data.

More recently, [Amato et al., 2024](#) proposed a method to cluster longitudinal ordinal data by assuming an underlying mixture of matrix-variate distributions.

Finally, looking at mixed-type longitudinal data, one main methodology to deal with such data lies in the framework of discrete (time-constant or varying) random intercepts for modeling heterogeneity, that includes mixture random effect models for longitudinal data extended to deal with multivariate and mixed outcomes by [Proust-Lima et al., 2013](#) and growth mixture models ([Ram et al., 2009](#)), where individuals are grouped in classes having a specific growth structure variability. These approaches are similar in that they model the change over time at both the population level and the individual level using random effects (or latent variables). In [Komarek et al., 2013](#) the authors rely on a multivariate extension of the classical generalized linear mixed model where a mixture distribution is additionally assumed for random effects. [Vávra et al., 2021](#) extend this model presenting a statistical model for joint modeling of mixed-type longitudinal data, while performing unsupervised clustering with respect to different covariate patterns. However, nominal (polytomous) variables are not taken into account in neither of the papers and time-dependent information is neglected. This work is expanded and improved in [Vvra et al., 2024](#). In [Cagnone et al., 2018](#) the authors extended the latent class model to take into account time evolution by means of latent Markov variable ([Bartolucci et al., 2019](#)) to model longitudinal binary and ordinal data on alcohol use disorder. In a model-based clustering perspective, [De la Cruz-Mesia et al., 2008](#) proposed a mixture of hierarchical nonlinear models for describing nonlinear relationships across time. [Manrique-Vallier, 2014](#) introduced a clustering strategy based on a mixed membership framework for analyzing discrete multivariate longitudinal data.

1.2 Preliminaries

Let $Z \sim \mathcal{MN}_{(J \times T)}(M, \Phi, \Sigma)$, that is a matrix-variate normal distribution where $M \in \mathbb{R}^{J \times T}$ is the matrix of means, $\Phi \in \mathbb{R}^{T \times T}$ is a covariance matrix containing the variances and covariances between the T occasions or times and $\Sigma \in \mathbb{R}^{J \times J}$ is the covariance matrix containing the variances and covariances of the J variables. The matrix-normal probability density function is given by

$$f(Z|M, \Phi, \Sigma) = (2\pi)^{-\frac{TJ}{2}} |\Phi|^{-\frac{J}{2}} |\Sigma|^{-\frac{T}{2}} \exp \left\{ -\frac{1}{2} \text{tr}[\Sigma^{-1}(Z - M)\Phi^{-1}(Z - M)^{\top}] \right\}. \quad (1)$$

The matrix-normal distribution represents a natural extension of the multivariate normal distribution, since if $Z \sim \mathcal{MN}_{(J \times T)}(M, \Phi, \Sigma)$, then $\text{vec}(Z) \sim \mathcal{MVN}_{JT}(\text{vec}(M), \Phi \otimes \Sigma)$, where $\text{vec}(\cdot)$ is the vectorization operator, that is the function mapping from a $J \times T$ matrix to a JT -dimensional vector, and \otimes denotes the Kronecker product. The property of rewriting the general covariance matrix $\Psi \in \mathbb{R}^{JT \times TJ}$ as $\Psi = \Phi \otimes \Sigma$ is called separability

condition. Then, the mean and the variance of the matrix-normal distribution are:

$$\mathbb{E}(\text{vec}(Z)|M, \Phi, \Sigma) = \text{vec}(M) \quad \text{and} \quad \mathbb{V}(\text{vec}(Z)|M, \Phi, \Sigma) = \Psi. \quad (2)$$

Being a special case of the multivariate normal distribution, the matrix-normal distribution shares the same properties, like, for instance, closure under marginalization, conditioning and linear transformations (Gupta et al., 2000). The separability condition of the covariance matrix has two advantages. First, it allows the modeling of the temporal pattern of interest directly on the covariance matrix Φ . Second, it represents a more parsimonious solution than that of the unrestricted Ψ .

Introduced by Violi, 2011a, the pdf of the finite Mixture of Matrix-Normals (MMN) model is given by

$$f(Z|\boldsymbol{\pi}, \boldsymbol{\Theta}) = \sum_{k=1}^K \pi_k \mathcal{MN}_{(J \times T)}(Z|M_k, \Phi_k, \Sigma_k),$$

where K is the number of mixture components, $\boldsymbol{\pi} = \{\pi_k\}_{k=1}^K$ is the vector of mixing proportions, subject to constraint $\sum_{k=1}^K \pi_k = 1$ and $\boldsymbol{\Theta} = \{\Theta_k\}_{k=1}^K$ is the set of component-specific parameters with $\Theta_k = \{M_k, \Phi_k, \Sigma_k\}$.

1.3 Our idea

As we aim to develop a model easily understandable and interpretable by practitioners with non-statistical background, we found matrix-variate distributions particularly fit, as shown in Alaimo et al., 2023. Moreover, as noticed in Anderlucci et al., 2015, the use of matrix-variate distributions allow to drop the conditional independence assumption, frequently implied in longitudinal latent variable models. Despite the efficacy of matrix-variate distributions, up to now these methods have only been applied to continuous data. We introduce a Mixture for Mixed Matrices (MMM) model, aiming at expanding the use to matrix-variate mixtures to ordinal data in an unsupervised learning context. Our model expands the use of matrix-variate mixtures to mixed-type data, by building on the framework proposed by McParland et al., 2016 and Choi et al., 2023 in the cross-sectional context.

In the following, in Sections 2 and 3 we will detail our model and the MCMC-EM algorithm to perform inference, respectively. In Section 4 some results on synthetic data are presented to assess the performance of the model. Finally, in Section 5 an real-world application concerning stock exchange data during the Covid-19 pandemic period is outlined.

2 The MMM model

Let denote by y_{ijt} the observation of the j -th ($j = 1, \dots, J$) variable for the i -th ($i = 1, \dots, N$) unit at time t ($t = 1, \dots, T$), that is: imagine to observe N units and measuring J different mixed variables T times throughout the course of the study. We can divide the J mixed variables into C continuous variables, O ordinal, binary and nominal ones and G as count variables, such that $C + O + G = J$. We are going to put ordinal, binary and nominal variables together as we will treat them in the same way.

Let us reorganize this data in a random-matrix form such that we denote the observed record of the i -th subject as $Y_i \in \mathbb{R}^{J \times T}$. $\mathbf{Y} = \{Y_i\}_{i=1}^N$ is a sample of $J \times T$ -variate matrix observations $Y_i \in [\mathbb{R}^{C \times T}, \mathbb{N}^{O \times T}, \mathbb{N}_0^{G \times T}]^\top$, $J = C + O + G$. The ordinal, binary and nominal classes are arbitrarily coded by non-negative integers such that each variable O has levels $\{1, 2, \dots, C_o\} \in \mathbb{N}$ ¹.

Then, we assume that each variable y_{ijt} is the manifestation of an underlying latent continuous variable z_{ijt} .

2.1 Modeling continuous variables

Let c indicate the generic c -th continuous variable. We assume that the observed continuous variables y_{ict} matches exactly the latent variable:

$$y_{ict} = z_{ict}$$

2.2 Modeling categorical ordinal variables

To map ordinal data, we follow [Amato et al., 2024](#). Let the generic ordinal o -th have C_o levels. Let γ_o denote a $C_o + 1$ -dimensional vector of thresholds that partition the real line for the corresponding o -th underlying continuous variable, and let the threshold parameters be constrained such that $-\infty = \gamma_{o,0} \leq \gamma_{o,1} \leq \dots \leq \gamma_{o,C_o} = \infty$. If the latent z_{iot} is such that $\gamma_{o,c-1} < z_{iot} < \gamma_{o,c}$ then the observed ordinal response, $y_{iot} = c$.

Moreover, let define $\mathcal{O}^{O \times T}$ the set of ordinal matrices of size $J \times T$ whose elements takes values in $\{1, \dots, C_o\}$. Each element of $\mathcal{O}^{O \times T}$ is called a response pattern. Let R be the cardinality of $\mathcal{O}^{O \times T}$. Each response pattern $Y_r \in \mathcal{O}^{O \times T}$ is generated by a portion Ω_r of the latent space $\mathbb{R}^{O \times T}$ according to thresholds $\boldsymbol{\gamma} := \{\gamma_o\}_{o=1}^O$. Let the binary vector $\tilde{Y}_i = (\tilde{Y}_{i1}, \dots, \tilde{Y}_{iR})$ be one-hot encoding of Y_i such that if the r -th pattern is observed then $\tilde{Y}_{ir} = 1$ and any other entry in the vector equals zero.

A key point is of course the choice of the thresholds $\boldsymbol{\gamma} = \{\gamma_j\}_{o=1}^O$. to avoid identifiability and computational complexity issues, thresholds are fixed and not considered as

¹In this work, we will consider zero not included in the set of natural numbers. We will use the notation \mathbb{N}_0 to indicate $\mathbb{N} \cup \{0\}$

parameters. There are different ways to do it. We decide to follow [Corneli et al., 2020](#), where the thresholds are chosen as $\gamma_o = (-\infty, 1.5, 2.5, \dots, C_o - 0.5, \infty)$.

2.3 Modeling categorical nominal variables

For categorical nominal data with P levels we can consider a one-hot encoding for $P - 1$ levels and treat them as binary variables. Binary variables can be considered as a special case of ordinal variables where the number of classes $C_o = 2$. The threshold cutting the underlying continuous variable is set to 0.

2.4 Modelling count variables

For count data we consider a Matrix variate Poisson-log normal distribution ([Silva et al., 2023](#)). Let g be the generic g -th count variable, then we assume that y_{igt} follows a Poisson distribution with parameter $\exp(z_{igt})$, where z_{igt} is a term of the $G \times T$ underlying latent matrix following a matrix normal distribution.

2.5 Joint model

So, we can think of Y_i as a block matrix, and conveniently split it between the first C rows, representing the observed continuous variables, followed by O rows representing the categorical variables and the remaining $J - C - O = G$ rows, representing the count variables. Notice that the slicing happens just over rows but not over columns. Then, we can write $Y_i = [Y_i^\alpha, Y_i^\beta, Y_i^\gamma]^\top$, where $Y_i^\alpha \in \mathbb{R}^{C \times T}$ is the block containing the continuous variables and $Y_i^\beta \in \mathbb{N}^{O \times T}$ gathers the categorical ones (that we coded via integers) and the binary ones, and $Y_i^\gamma \in \mathbb{N}_0^{G \times T}$ is the block containing the count variables.

At this point, we can assume that each observed block of the matrix Y_i is indeed the manifestation of the corresponding block of the latent random matrix $Z_i = [Z_i^\alpha, Z_i^\beta, Z_i^\gamma]^\top$, and that this underlying random matrix is linked through different relations to the observed matrix Y_i , depending on the type of variable each element y_{ijt} , as described previously.

Then, we assume a mixture of matrix-normal distributions on the latent space. We can consequently write

$$\begin{pmatrix} Z_i^\alpha \\ Z_i^\beta \\ Z_i^\gamma \end{pmatrix} \sim \sum_{k=1}^K \pi_k \mathcal{MN}_{(J \times T)} \left(\begin{pmatrix} M_k^\alpha \\ M_k^\beta \\ M_k^\gamma \end{pmatrix}, \Phi_k, \begin{pmatrix} \Sigma_k^{\alpha\alpha} & \Sigma_k^{\alpha\beta} & \Sigma_k^{\alpha\gamma} \\ \Sigma_k^{\beta\alpha} & \Sigma_k^{\beta\beta} & \Sigma_k^{\beta\gamma} \\ \Sigma_k^{\gamma\alpha} & \Sigma_k^{\gamma\beta} & \Sigma_k^{\gamma\gamma} \end{pmatrix} \right). \quad (3)$$

From here, we can derive the joint model. To keep notation coherent, let define with \tilde{Y}_i^β the one-hot encoding of the categorical part of Y_i as described in Section 2.2. In addition to Z_i , we introduce a latent binary K -dimensional allocation vector that indicate

whether the unit i belongs to the k -th cluster, $\ell_i = (\ell_{i1}, \dots, \ell_{iK})$, such that $\ell_{ik} = 1$ if the i -th unit belongs to the k -th cluster.

Recalling the links each kind of observed variables has with the latent ones, we can express our model through the following distributional assumptions:

$$\begin{aligned}
\ell_i &\sim \mathcal{M}(1, \boldsymbol{\pi}), \quad \boldsymbol{\pi} := (\pi_1, \dots, \pi_K) \\
Z_i^\alpha | \ell_{ik} = 1 &\sim \mathcal{MN}_{(C \times T)}(Z_i^\alpha | \Theta_k^\alpha), \quad \Theta_k^\alpha := \{M_k^\alpha, \Phi_k, \Sigma_k^\alpha\}, \\
Z_i^\beta | Z_i^\alpha, \ell_{ik} = 1 &\sim \mathcal{MN}_{(O \times T)}(Z_i^\beta | \Theta_k^{\beta|\alpha}), \quad \Theta_k^{\beta|\alpha} := \{M_k^{\beta|\alpha}, \Phi_k, \Sigma_k^{\beta|\alpha}\}, \\
Z_i^\gamma | Z_i^\alpha, Z_i^\beta, \ell_{ik} = 1 &\sim \mathcal{MN}_{(G \times T)}(Z_i^\gamma | \Theta_k^{\gamma|\alpha, \beta}), \quad \Theta_k^{\gamma|\alpha, \beta} := \{M_k^{\gamma|\alpha, \beta}, \Phi_k, \Sigma_k^{\gamma|\alpha, \beta}\}; \\
\tilde{Y}_i^\beta | Z_i^\beta, \ell_{ik} = 1 &\sim \mathcal{M}(1, \xi_i), \quad \xi_i := (\mathbf{1}_{\Omega_1}(Z_i^\beta), \dots, \mathbf{1}_{\Omega_R}(Z_i^\beta)), \\
Y_{igt}^\gamma | Z_{igt}^\gamma &\sim \mathcal{P}(\exp(Z_{igt}^\gamma)),
\end{aligned}$$

where \mathcal{M} indicates the multinomial distribution and $\mathbf{1}_{\Omega_r}(Z_i^\beta)$ is the indicator function that equals 1 when the elements in Z_i^β have values that determine the r -th pattern. Hence, when $\tilde{Y}_{ir}^\beta = 1$, the vector ξ_i is a vector whose r -th element equals 1 and all the others equal 0.

Further, to avoid assuming the independence between the different blocks, to link the matrix latent distributions we resort to condition on one block to another by using the properties of matrix-variate normal distribution (Gupta et al., 2000). Thus, $\Theta_k^{\gamma|\alpha, \beta} := \{M_k^{\gamma|\alpha, \beta}, \Phi_k, \Sigma_k^{\gamma|\alpha, \beta}\}$, more precisely $M_k^{\gamma|\alpha, \beta} = M_k^\gamma + \Sigma_k^\gamma \Sigma_k^{-1, \cdot \cdot} (Z_i^{\alpha, \beta} - M_k^{\alpha, \beta})$ and $\Sigma_k^{\gamma|\alpha, \beta} = \Sigma_k^{\gamma\gamma} - \Sigma_k^\gamma \Sigma_k^{-1, \cdot \cdot} \Sigma_k^{\gamma\alpha}$, and where $\Theta_k^{\beta|\alpha} := \{M_k^{\beta|\alpha}, \Phi_k, \Sigma_k^{\beta|\alpha}\}$, more precisely $M_k^{\beta|\alpha} = M_k^\beta + \Sigma_k^{\beta\alpha} \Sigma_k^{-1, \alpha\alpha} (Y_i^\alpha - M_k^\alpha)$ and $\Sigma_k^{\beta|\alpha} = \Sigma_k^{\beta\beta} - \Sigma_k^{\beta\alpha} \Sigma_k^{-1, \alpha\alpha} \Sigma_k^{\alpha\beta}$.

Lastly, Assuming that the observed value pattern of \tilde{Y}_i^β is r for sake of notation, we can compose the distribution of each observed mixed matrix as

$$\begin{aligned}
Y_i &\sim \sum_{k=1}^K \pi_k \mathcal{MN}_{(C \times T)}(Z_i^\alpha | \Theta_k^\alpha) \cdot \int_{\Omega_r} \mathcal{MN}_{(O \times T)}(Z_i^\beta | \Theta_k^{\beta|\alpha}) dZ_i^\beta \\
&\cdot \int_{\mathbb{R}} \prod_t^T \prod_g^G \mathcal{P}(y_{igt}^\gamma | \exp(z_{igt}^\gamma)) \cdot \mathcal{MN}_{(G \times T)}(Z_i^\gamma | \Theta_k^{\gamma|\alpha, \beta}) dZ_i^\gamma. \tag{4}
\end{aligned}$$

2.6 Likelihood

In the following, $\mathbf{Z} := \{Z_i\}_{i=1}^N$, $\boldsymbol{\ell} := \{\ell_i\}_{i=1}^N$ will indicate the ensembles of Z_i and ℓ_i respectively, and $\mathbf{Y} := \{Y_i\}_{i=1}^N$ be the collection of the observed matrices Y_i . Finally, the set of unknown parameters to be estimated is $\boldsymbol{\Theta} := \{\pi_k, M_k, \Phi_k, \Sigma_k\}_{k=1}^K$.

The joint density of $Y_i^\gamma, Z_i^\gamma, \tilde{Y}_i^\beta, Z_i^\beta, Z_i^\alpha, \ell_i$ is:

$$f(Y_i^\gamma, Z_i^\gamma, \tilde{Y}_i^\beta, Z_i^\beta, Z_i^\alpha, \ell_i) = f(Y_i^\gamma | Z_i^\gamma, \tilde{Y}_i^\beta, Z_i^\beta, Z_i^\alpha, \ell_i) \cdot f(Z_i^\gamma | \tilde{Y}_i^\beta, Z_i^\beta, Z_i^\alpha, \ell_i) \cdot f(\tilde{Y}_i^\beta | Z_i^\beta, Z_i^\alpha, \ell_i) \cdot f(Z_i^\beta | Z_i^\alpha, \ell_i) \cdot f(\ell_i).$$

We can therefore write the complete log-likelihood as:

$$\begin{aligned} \mathcal{L}_C(\Theta; \mathbf{Y}, \mathbf{Z}, \ell) &= \prod_{i=1}^N \prod_{k=1}^K \left[\pi_k \cdot \left(\prod_t^T \prod_g^G \mathcal{P}(y_{igt}^\gamma | \exp(z_{igt}^\gamma)) \right) \cdot \mathcal{MN}_{(G \times T)}(Z_i^\gamma | \Theta_k^{\gamma|\alpha, \beta}) \cdot \right. \\ &\quad \left. \mathcal{MN}_{(O \times T)}(Z_i^\beta | \Theta_k^{\beta|\alpha}) \cdot \mathcal{MN}_{(C \times T)}(Z_i^\alpha | \Theta_k^\alpha) \cdot \prod_{r=1}^R \mathbf{1}_{\Omega_r}(Z_i^\beta)^{\tilde{Y}_{ir}^\beta} \right]^{\ell_{ik}}. \end{aligned} \quad (5)$$

We acknowledge the identity $Y_i^\alpha = Z_i^\alpha$ and the fact that the last term is non-stochastic since conditioning on \mathbf{Z} implies that the value of Z_i^β is known and we can discard it. Therefore, by using the notation of Equation 3, we can rewrite this equation compactly as the complete log-likelihood can be written as:

$$\begin{aligned} \log \mathcal{L}_C(\Theta; \mathbf{Y}, \mathbf{Z}, \ell) &= \sum_{i=1}^N \sum_{k=1}^K \ell_{ik} \left[C + \log(\pi_k) - \frac{J}{2} \log(|\Phi_k|) - \frac{T}{2} \log(|\Sigma_k|) - \right. \\ &\quad \left. \frac{1}{2} \text{tr}[\Sigma_k^{-1}(Z_i - M_k)\Phi_k^{-1}(Z_i - M_k)^\top] \right]. \end{aligned} \quad (6)$$

where C is a constant with respect to the set of parameters Θ .

On the other hand, we can define the observed likelihood as $\mathcal{L}_O(\Theta; \mathbf{Y})$, that is:

$$\begin{aligned} \mathcal{L}_O(\Theta; \mathbf{Y}) &:= \prod_{i=1}^N \left\{ \sum_{k=1}^K \pi_k \mathcal{MN}_{(C \times T)}(Z_i^\alpha | \Theta_k^\alpha) \cdot \int_{\Omega_r} \mathcal{MN}_{(O \times T)}(Z_i^\beta | \Theta_k^{\beta|\alpha}) dZ_i^\beta \right. \\ &\quad \left. \cdot \int_{\mathbb{R}} \prod_t^T \prod_g^G \mathcal{P}(y_{igt}^\gamma | \exp(z_{igt}^\gamma)) \times \mathcal{MN}_{(G \times T)}(Z_i^\gamma | \Theta_k^{\gamma|\alpha, \beta}) dZ_i^\gamma \right\}. \end{aligned} \quad (7)$$

3 Inference

In our model, we are assuming two different latent (unobserved) variables. Therefore, we will use the Expectation-Maximization (EM) algorithm (Dempster et al., 1977) to infer the MMM model's parameters. The EM algorithm is well-suited for situations involving latent variables or unobserved data, as it allows for the estimation of model parameters despite the incompleteness.

3.1 EM-algorithm

The EM algorithm is an iterative algorithm that alternates two steps: the expectation step (E-step) and the maximization step (M-step). It starts from an initialization $\hat{\Theta}^{(0)}$ of the parameters. Then, let denote with the superscript $(s + 1)$ the parameters estimated in the current step and with (s) the ones computed in the previous step.

For the MMM model, the E-step consists of evaluating $Q(\Theta, \hat{\Theta}^{(s)}) := \mathbb{E}(\log \mathcal{L}_C(\Theta; \mathbf{Y}, \mathbf{Z}, \ell) | \hat{\Theta}^{(s)}, \mathbf{Y})$, that is the expectation of the complete log-likelihood conditioned on the parameters computed in the previous step and on the observed data. In the M-step the parameters are updated by maximizing the expected log-likelihood found on the E step, that is $\hat{\Theta}^{(s+1)} := \arg \max_{\Theta} Q(\Theta, \hat{\Theta}^{(s)})$. The iteration process is repeated until convergence on the log-likelihood is met.

3.2 Initialization

To find the initial values of $\hat{\Theta}^{(0)}$ mentioned in Section 3.1, our proposal is the following. Identity matrices are chosen for the initialization of the covariance matrices Φ_k and Σ_k . For the initialization of M_k and π_k , two solutions are proposed and tested in Section 4.3. The first is a Kmeans++ (Arthur et al., 2007) initialization, that is performed on the vectorized data. The second is a multiple random initialization: the mean matrices M_k are chosen by uniform sampling K matrices among the N observed data matrices. Since the EM algorithm is not guaranteed to converge toward a global optimum, the algorithm is applied multiple times and the results with the highest log-likelihood is selected. For simulations in Section 4.3, 5 random initializations proved to be enough, but a higher number might be needed for more complex settings.

Both the initialization techniques are applied on the latent space, meaning that for count data they are applied on the logarithm of the observed data.

3.3 E-step

As previously stated, the E-step consists of evaluating $Q(\Theta, \hat{\Theta}^{(s)}) := \mathbb{E}(\log \mathcal{L}_C(\Theta; \mathbf{Y}, \mathbf{Z}, \ell) | \hat{\Theta}^{(s)}, \mathbf{Y})$, that is the expectation of the complete log-likelihood conditioned on the parameters computed in the previous step and on the observed data.

We can expand Equation 6 as:

$$\log \mathcal{L}_C(\Theta; \mathbf{Y}, \mathbf{Z}, \ell) = \sum_{i=1}^N \sum_{k=1}^K \ell_{ik} \left[C + \log(\pi_k) - \frac{J}{2} \log(|\Phi_k|) - \frac{T}{2} \log(|\Sigma_k|) - \frac{1}{2} \text{tr}[\Sigma_k^{-1} Z_i \Phi_k^{-1} Z_i^\top - \Sigma_k^{-1} Z_i \Phi_k^{-1} M_k^\top - \Sigma_k^{-1} M_k \Phi_k^{-1} Z_i^\top + \Sigma_k^{-1} M_k \Phi_k^{-1} M_k^\top] \right]. \quad (8)$$

Then, from Equation 8, it is easy to see that the expected values to be computed are $\mathbb{E}(\ell_{ik} | \hat{\Theta}^{(s)}, \mathbf{Y})$, $\mathbb{E}(\ell_{ik} Z_i | \hat{\Theta}^{(s)}, \mathbf{Y})$ and $\mathbb{E}(\ell_{ik} Z_i \Phi_k^{-1(s)} Z_i^\top | \hat{\Theta}^{(s)}, \mathbf{Y})$ or $\mathbb{E}(\ell_{ik} Z_i^\top \Sigma_k^{-1(s)} Z_i | \hat{\Theta}^{(s)}, \mathbf{Y})$ by the cyclic property of the trace. As we will see in Section 3.4, we will need both.

We will proceed with their computation one by one. First, $\mathbb{E}(\ell_{ik} | \hat{\Theta}^{(s)}, \mathbf{Y})$ can be computed according to the Bayes' rule as

$$\mathbb{E}(\ell_{ik} | \hat{\Theta}^{(s)}, \mathbf{Y}) = \frac{q_{ik}}{\sum_{h=1}^K q_{ih}} =: \hat{\tau}_{ik}^{(s+1)} \quad (9)$$

where

$$q_{ik} = \pi_k \mathcal{MN}_{(C \times T)}(Z_i^\alpha | \Theta_k^{(s), \alpha}) \cdot \int_{\Omega_r} \mathcal{MN}_{(O \times T)}(Z_i^\beta | \Theta_k^{(s), \beta | \alpha}) dZ_i^\beta \cdot \int_{\mathbb{R}} \prod_t^T \prod_g^G \mathcal{P}(y_{igt}^\gamma | \exp(z_{igt}^\gamma)) \cdot \mathcal{MN}_{(G \times T)}(Z_i^\gamma | \Theta_k^{\gamma | \alpha, \beta}) dZ_i^\gamma$$

where the first integral can be approximated through a Monte-Carlo approach applied on the vectorized reparametrization of the matrix-variate distribution and the second one can be approximated by using the estimated value for Z_i^γ presented in the following.

For $\mathbb{E}(\ell_{ik} Z_i | \hat{\Theta}^{(s)}, \mathbf{Y})$, recalling the block structure of Z_i , we can write

$$\begin{aligned} \mathbb{E}(\ell_{ik} Z_i | \hat{\Theta}^{(s)}, \mathbf{Y}) &= \mathbb{P}(\ell_{ik} = 1 | \hat{\Theta}^{(s)}, \mathbf{Y}) \cdot \mathbb{E} \left(\begin{bmatrix} Z_i^\alpha \\ Z_i^\beta \\ Z_i^\gamma \end{bmatrix} \middle| \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y} \right) = \\ &= \mathbb{P}(\ell_{ik} = 1 | \hat{\Theta}^{(s)}, \mathbf{Y}) \cdot \begin{bmatrix} Y_i^\alpha \\ \mathbb{E}(Z_i^\beta | M_k^{\beta | \alpha, (s)}, \Phi_k^{(s)}, \Sigma_k^{\beta | \alpha, (s)}) \\ \mathbb{E}(Z_i^\gamma | M_k^{\gamma | \alpha, \beta, (s)}, \Phi_k^{(s)}, \Sigma_k^{\gamma | \alpha, \beta, (s)}) \end{bmatrix} := \hat{\tau}_{ik}^{(s+1)} \cdot \begin{bmatrix} Y_i^\alpha \\ \hat{M}_{ik}^{\beta, (s+1)} \\ \hat{M}_{ik}^{\gamma, (s+1)} \end{bmatrix}, \quad (10) \end{aligned}$$

where the matrix-variate expectation related to count data can be computed by defining $z_i^\gamma \in \mathbb{R}^{GT \times 1}$ as the vectorized version of Z_i^γ and computing its expectation

$\hat{m}_{ik}^{\gamma, (s+1)} := \mathbb{E}(z_i^\gamma | \ell_{ik} = 1, \mathbf{Y}, \hat{\Theta}^{(s)})$ by means of the sampler implemented in the R package **Rstan**, that is the R interface to the **Stan** software (Stan Development Team, 2024).

The matrix-variate expectation related to categorical data can be computed by defining $z_i^\beta \in \mathbb{R}^{OT \times 1}$ as the vectorized version of Z_i^β and computing its expectation $\hat{m}_{ik}^{\beta, (s+1)} := \mathbb{E}(z_i^\beta | \ell_{ik} = 1, \mathbf{Y}, \hat{\Theta}^{(s)})$ through the use of a Gibbs sampler to sample from a truncated multivariate normal distribution.

Finally, for $\mathbb{E}(\ell_{ik} Z_i \Phi_k^{-1} Z_i^\top | \hat{\Theta}^{(s)}, \mathbf{Y})$ we have:

$$\begin{aligned} \mathbb{E}(\ell_{ik} Z_i \Phi_k^{-1} Z_i^\top | \hat{\Theta}^{(s)}, \mathbf{Y}) &= \mathbb{P}(\ell_{ik} = 1 | \hat{\Theta}^{(s)}, \mathbf{Y}) \cdot \mathbb{E}(Z_i \Phi_k^{-1} Z_i^\top | \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y}) = \\ &= \hat{\tau}_{ik}^{(s+1)} \cdot \begin{bmatrix} Y_i^\alpha \hat{\Phi}_k^{-1(s)} Y_i^{\alpha\top} & Y_i^\alpha \hat{\Phi}_k^{-1(s)} \hat{M}_{ik}^{\beta, \tau(s+1)} & Y_i^\alpha \hat{\Phi}_k^{-1(s)} \hat{M}_{ik}^{\gamma, \tau(s+1)} \\ \hat{M}_{ik}^{\beta, (s+1)} \hat{\Phi}_k^{-1(s)} Y_i^{\alpha\top} & \hat{D}_{ik}^{(s+1)} & \hat{M}_{ik}^{\beta, (s+1)} \hat{\Phi}_k^{-1(s)} \hat{M}_{ik}^{\gamma, \tau(s+1)} \\ \hat{M}_{ik}^{\gamma, (s+1)} \hat{\Phi}_k^{-1(s)} Y_i^{\alpha\top} & \hat{M}_{ik}^{\gamma, (s+1)} \hat{\Phi}_k^{-1(s)} \hat{M}_{ik}^{\beta, \tau(s+1)} & \hat{B}_{ik}^{(s+1)} \end{bmatrix}, \end{aligned} \quad (11)$$

where $\hat{D}_{ik}^{(s+1)} := \mathbb{E}(Z_i^\beta \Phi_k^{-1} Z_i^{\beta\top} | \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y})$ and $B_{ik}^{(s+1)} := \mathbb{E}(Z_i^\gamma \Phi_k^{-1} Z_i^{\gamma\top} | \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y})$.

To compute $D_{ik}^{(s+1)}$ we make use of the the elements of $\hat{S}_{ik}^{\beta, (s+1)} := \mathbb{E}(z_i^\beta z_i^{\beta\top} | \ell_{ik} = 1, \mathbf{Y}, \hat{\Theta}^{(s)})$. The samples generated to calculate the first moment $\hat{m}_{ik}^{\beta, (s+1)}$ can be reused to compute the matrix $\hat{S}_{ik}^{\beta, (s+1)}$ by calculating the mean of the inner product between them. Similarly, for $\hat{B}_{ik}^{(s+1)}$, we make use of the the elements of $\hat{S}_{ik}^{\gamma, (s+1)} := \mathbb{E}(z_i^\gamma z_i^{\gamma\top} | \ell_{ik} = 1, \mathbf{Y}, \hat{\Theta}^{(s)})$. As before, the samples generated to calculate the first moment $\hat{m}_{ik}^{\gamma, (s+1)}$ can be reused to compute the matrix $\hat{S}_{ik}^{\gamma, (s+1)}$.

On the other hand, to compute $\mathbb{E}(\ell_{ik} Z_i^\top \Sigma_k^{-1} Z_i | \hat{\Theta}^{(s)}, \mathbf{Y})$:

$$\begin{aligned} \mathbb{E}(\ell_{ik} Z_i^\top \Sigma_k^{-1} Z_i | \hat{\Theta}^{(s)}, \mathbf{Y}) &= \mathbb{P}(\ell_{ik} = 1 | \hat{\Theta}^{(s)}, \mathbf{Y}) \cdot \mathbb{E}(Z_i^\top \Sigma_k^{-1} Z_i | \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y}) = \\ &= \hat{\tau}_{ik}^{(s+1)} \cdot \left(Y_i^{\alpha\top} \hat{\Sigma}_k^{-1, \alpha\alpha} Y_i^\alpha + Y_i^{\alpha\top} \hat{\Sigma}_k^{-1, \alpha\beta} \hat{M}_{ik}^{\beta, (s+1)} + Y_i^{\alpha\top} \hat{\Sigma}_k^{-1, \alpha\gamma} \hat{M}_{ik}^{\gamma, (s+1)} + \right. \\ &\quad \hat{M}_{ik}^{\beta, (s+1)\top} \hat{\Sigma}_k^{-1, \beta\alpha} Y_i^\alpha + \hat{C}_{ik}^{(s+1)} + \hat{M}_{ik}^{\beta, (s+1)\top} \hat{\Sigma}_k^{-1, \beta\gamma} \hat{M}_{ik}^{\gamma, (s+1)} + \\ &\quad \left. \hat{M}_{ik}^{\gamma, (s+1)\top} \hat{\Sigma}_k^{-1, \gamma\alpha} Y_i^\alpha + \hat{M}_{ik}^{\gamma, (s+1)\top} \hat{\Sigma}_k^{-1, \gamma\beta} \hat{M}_{ik}^{\beta, (s+1)} + \hat{A}_{ik}^{(s+1)} \right), \end{aligned} \quad (12)$$

where $\hat{C}_{ik}^{(s+1)} := \mathbb{E}(Z_i^{\beta\top} \Sigma_k^{\beta\beta} Z_i^\beta | \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y})$, $\hat{A}_{ik}^{(s+1)} := \mathbb{E}(Z_i^{\gamma\top} \Sigma_k^{\gamma\gamma} Z_i^\gamma | \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y})$ and $\hat{\Sigma}_k^{-1, **}$ indicated the corresponding block of the inverted matrix $\hat{\Sigma}_k^{-1}$ with respect to the notation in Equation 3. Again, to compute $\hat{C}_{ik}^{(s)}$ we will make use of the elements of

$\hat{S}_{ik}^{\beta,(s+1)}$, while for $\hat{A}_{ik}^{(s)}$ we will use the elements of $\hat{S}_{ik}^{\gamma,(s+1)}$.

Summing up, this means that computing $\mathbb{E}(\log \mathcal{L}_C(\Theta; \mathbf{Y}, \mathbf{Z}, \ell) | \hat{\Theta}^{(s)}, \mathbf{Y})$ requires to compute:

- $\mathbb{E}(\ell_{ik} | \mathbf{Y}, \hat{\Theta}^{(s)}) =: \hat{\tau}_{ik}^{(s+1)}$,
- $\mathbb{E}(z_i^\beta | \ell_{ik} = 1, \mathbf{Y}, \hat{\Theta}^{(s)}) =: \hat{m}_{ik}^{\beta,(s+1)}$,
- $\mathbb{E}(z_i^\beta z_i^{\beta\top} | \ell_{ik}, \mathbf{Y}, \hat{\Theta}^{(s)}) =: \hat{S}_{ik}^{\beta,(s+1)}$, whose elements are required for the computation of $\hat{D}_{ik}^{(s+1)}$ and $\hat{C}_{ik}^{(s+1)}$,
- $\mathbb{E}(z_i^\gamma | \ell_{ik} = 1, \mathbf{Y}, \hat{\Theta}^{(s)}) =: \hat{m}_{ik}^{\gamma,(s+1)}$,
- $\mathbb{E}(z_i^\gamma z_i^{\gamma\top} | \ell_{ik}, \mathbf{Y}, \hat{\Theta}^{(s)}) =: \hat{S}_{ik}^{\gamma,(s+1)}$, whose elements are required for the computation of $\hat{B}_{ik}^{(s+1)}$ and $\hat{A}_{ik}^{(s+1)}$.

3.4 M-step

The updated for the parameters at step $(s+1)$ are given by

$$\hat{\pi}_k^{(s+1)} = \frac{\sum_{i=1}^N \hat{\tau}_{ik}^{(s+1)}}{N}, \quad \hat{M}_k^{(s+1)} = \frac{1}{\sum_{i=1}^N \hat{\tau}_{ik}^{(s+1)}} \sum_{i=1}^N \hat{\tau}_{ik}^{(s+1)} \begin{bmatrix} Y_i^\alpha \\ \hat{M}_{ik}^{\beta,(s+1)} \\ \hat{M}_{ik}^{\gamma,(s+1)} \end{bmatrix}, \quad (13)$$

$$\begin{aligned} \hat{\Sigma}_k^{(s+1)} = & \frac{1}{T \sum_{i=1}^N \hat{\tau}_{ik}^{(s+1)}} \sum_{i=1}^N \hat{\tau}_{ik}^{(s+1)} \times \\ & \left(\begin{bmatrix} Y_i^\alpha \hat{\Phi}_k^{-1(s)} Y_i^{\alpha\top} & Y_i^\alpha \hat{\Phi}_k^{-1(s)} \hat{M}_{ik}^{\beta,\top(s+1)} & Y_i^\alpha \hat{\Phi}_k^{-1(s)} \hat{M}_{ik}^{\gamma,\top(s+1)} \\ \hat{M}_{ik}^{\beta,(s+1)} \hat{\Phi}_k^{-1(s)} Y_i^{\alpha\top} & \hat{D}_{ik}^{(s+1)} & \hat{M}_{ik}^{\beta,(s+1)} \hat{\Phi}_k^{-1(s)} \hat{M}_{ik}^{\gamma,\top(s+1)} \\ \hat{M}_{ik}^{\gamma,(s+1)} \hat{\Phi}_k^{-1(s)} Y_i^{\alpha\top} & \hat{M}_{ik}^{\gamma,(s+1)} \hat{\Phi}_k^{-1(s)} \hat{M}_{ik}^{\beta,\top(s+1)} & \hat{B}_{ik}^{(s+1)} \end{bmatrix} - \right. \\ & \left. \hat{M}_k^{(s+1)} \hat{\Phi}_k^{-1(s)} \begin{bmatrix} Y_i^\alpha \\ \hat{M}_{ik}^{\beta,(s+1)} \\ \hat{M}_{ik}^{\gamma,(s+1)} \end{bmatrix}^\top - \begin{bmatrix} Y_i^\alpha \\ \hat{M}_{ik}^{\beta,(s+1)} \\ \hat{M}_{ik}^{\gamma,(s+1)} \end{bmatrix} \hat{\Phi}_k^{-1(s)} \hat{M}_k^{\top(s+1)} + \hat{M}_k^{(s+1)} \hat{\Phi}_k^{-1(s)} \hat{M}_k^{\top(s+1)} \right) \end{aligned} \quad (14)$$

The update formulas of the two covariance matrices are interconnected:

$$\begin{aligned}
\hat{\Phi}_k^{(s+1)} = & \frac{1}{J \sum_{i=1}^N \hat{\tau}_{ik}^{(s+1)}} \sum_{i=1}^N \hat{\tau}_{ik}^{(s+1)} \left(Y_i^{\alpha \top} \hat{\Sigma}_k^{-1, \alpha \alpha} Y_i^\alpha + Y^{\alpha \top} \hat{\Sigma}_k^{-1, \alpha \beta} \hat{M}_{ik}^{\beta, (s+1)} + Y^{\alpha \top} \hat{\Sigma}_k^{-1, \alpha \gamma} \hat{M}_{ik}^{\gamma, (s+1)} + \right. \\
& \hat{M}_{ik}^{\beta, (s+1) \top} \hat{\Sigma}_k^{-1, \beta \alpha} Y_i^\alpha + \hat{C}_{ik}^{(s+1)} + \hat{M}_{ik}^{\beta, (s+1) \top} \hat{\Sigma}_k^{-1, \beta \gamma} \hat{M}_{ik}^{\gamma, (s+1)} + \\
& \hat{M}_{ik}^{\gamma, (s+1) \top} \hat{\Sigma}_k^{-1, \gamma \alpha} Y_i^\alpha + \hat{M}_{ik}^{\gamma, (s+1) \top} \hat{\Sigma}_k^{-1, \gamma \beta} \hat{M}_{ik}^{\beta, (s+1)} + \hat{A}_{ik}^{(s+1)} - \\
& \hat{M}_k^{\top (s+1)} \hat{\Sigma}_k^{-1 (s+1)} \begin{bmatrix} Y_i^\alpha \\ \hat{M}_{ik}^{\beta, (s+1)} \\ \hat{M}_{ik}^{\gamma, (s+1)} \end{bmatrix} - \begin{bmatrix} Y_i^\alpha \\ \hat{M}_{ik}^{\beta, (s+1)} \\ \hat{M}_{ik}^{\gamma, (s+1)} \end{bmatrix}^\top \hat{\Sigma}_k^{-1 (s+1)} \hat{M}_k^{(s+1)} + \\
& \left. \hat{M}_k^{\top (s+1)} \hat{\Sigma}_k^{-1 (s+1)} \hat{M}_k^{(s+1)} \right) \tag{15}
\end{aligned}$$

3.5 Convergence

Because of the MCMC use during the E-step, the property of monotone increase of the observed log-likelihood does not hold for our model (McLachlan et al., 2008, Ruth, 2024). Therefore, to assess convergence we use moving average estimation on the observed log-likelihood.

Let l_o^s the observed log-likelihood at step s , then our convergence criterion is

$$\left| \frac{\left(\frac{1}{w_1} \sum_{i=s-w_1+1}^s l_o^i \right) - \left(\frac{1}{w_2} \sum_{i=s-w_1-w_2+1}^{s-w_1} l_o^i \right)}{\frac{1}{w_2} \sum_{i=s-w_1-w_2+1}^{s-w_1} l_o^i} \right| < \varepsilon.$$

In the following, $\varepsilon = 1 \cdot 10^{-3}$ is chosen.

3.6 Selection of the number of cluster K

The number of cluster K is selected by minimizing the BIC (Schwarz, 1978) criterion. The BIC for a number of cluster k is defined as

$$\text{BIC}_k := -2 \log \mathcal{L}_O(\Theta; \mathbf{Y}) + \nu_k \log(N),$$

where ν_k is the total number of model parameters:

$$\nu_k := k[1 + JT + J(J+1)/2 + T(T+1)/2] - 1,$$

and $\mathcal{L}_O(\Theta; \mathbf{Y})$ is the observed likelihood defined in Equation 7.

To select the model with the optimal K , the algorithm needs to be executed for every $k = 1, \dots, K$ and the model with the lowest BIC_k is chosen.

4 Simulation study

This section presents numerical experiments on simulated data in order to illustrate the behavior of the proposed model. First, we aim at studying the influence of the initialization procedure and sample size in estimating the partition and the parameters. Secondly, the robustness to different noise ratio in the data concerning the clustering, the parameters estimation and the model selection. Finally, we compare the MMM model to its continuous counterpart (MMN) when used on mixed data treated like continuous data.

4.1 Simulation Setup

20 different samples have been simulated for increasing number of units $N \in \{100, 500, 1000\}$, with number of clusters $K = 2$, number of variables $J = 4$, number of times $T = 3$ and cluster proportions $\pi = (0.6, 0.4)$. The J variables are of mixed type, with the first variable being continuous, the second being ordinal, the third being binary and the fourth being a counting variable. The ordinal variable has 5 levels. Each sample has been drawn from a matrix-variate Gaussian and then transformed according to the model described in Section 2. The distributions parameters were chosen as following: identity matrices for the covariance matrices Φ_k and Σ_k for each cluster, while mean matrices M_k chosen such that the estimated the optimal Adjusted Rand Index (ARI) (Rand, 1971), computed by performing one step of the clustering algorithm using the true parameters, would be around 0.85. This setting led to the choice of two mean matrices as described in Table C1.

Moreover, three scenarios are derived from this setting by adding some noise by adding to the underlying continuous latent matrix of a percentage τ of units a reasonable level of noise, generated according to a centered Gaussian with variance equal to 0.5, allocated to the two clusters proportionally to the clusters' size: 0% (scenario 1), 10% (scenario 2), 20% (scenario 3). The two different kinds of initialization described in Section 3.2 have been tested. Regarding the algorithm setup, we set to 100 iterations as the burn-in period of Gibbs sampler in the E-step, and a thinning equal to 2 to prevent too correlated samples. The number of simulated samples is set to 100. Concerning the simulation done via `stan`, we set the chain iterations to 500, of which half as burn-in, for 3 different chains.

4.2 Computational time

Computation time for one iteration on 2.40 GHz 11th Gen Intel Core i5-1135G7 with 16 Go RAM for one step of the algorithm with Kmeans++ initialization for $K = 1$ is about 5 seconds for $N = 100$ and about 30 seconds second for $N = 1000$.

4.3 Influence of initialization & sample size

We first aim at studying the ability of the algorithm to recover the simulated model depending on the type of initialization of the EM algorithm and on the size of the sample. Figure 1 shows the quality of estimated partitions assessed by means of ARI. We recall that an ARI of 1 indicates that the partition provided by the algorithm is perfectly aligned with the simulated one. Conversely, an ARI of 0 indicates that the two partitions could as well be some random matches. On the graph, the optimal ARI (≈ 0.85) according to the simulation scheme is represented by a horizontal line. The boxplots show some small differences in the median values of the ARI measurements between the two initialization methods, with the random multistart initialization performing moderately better than its Kmeans++ counterpart, both in terms of median and of lower variability. When the sample size is sufficiently large, the result that stems from the multistart initialization almost attains the optimal ARI.

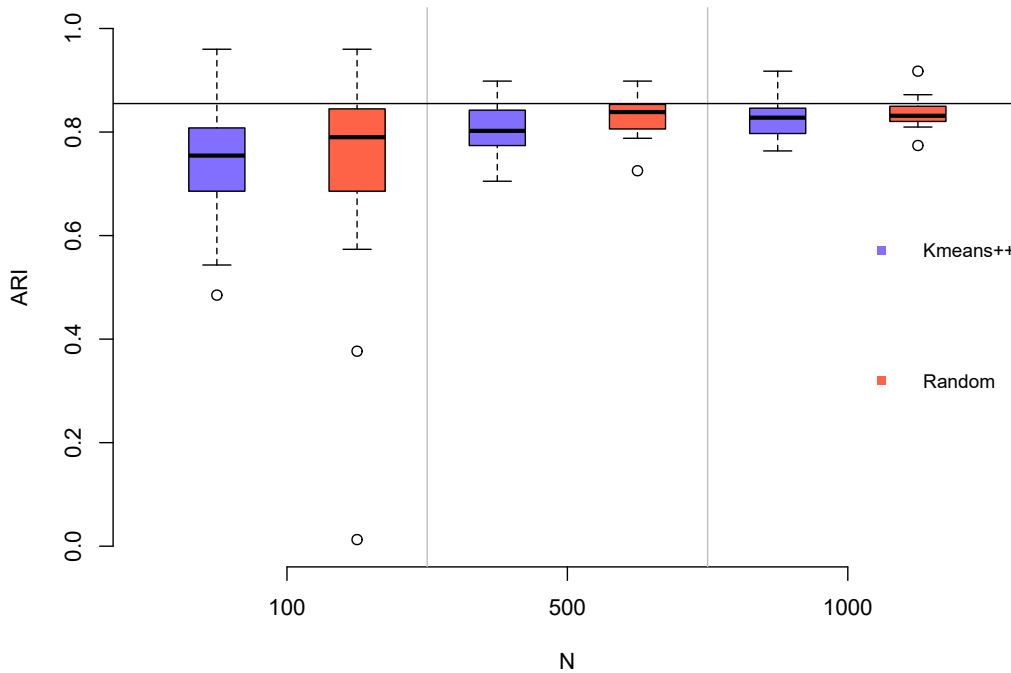


Figure 1: Influence of initialization and sample size. The horizontal line represents the estimated Bayesian error.

However, while the random multistart initialization seems to perform marginally better than the Kmeans++ from a partitioning point of view, it is noteworthy to consider

the computational and temporal burden of the former compared to the latter. One might consider whether the trade-off is worthy on a case-by-case base.

In addition, we measure their performance also by computing the Mean Absolute Percentage Error (MAPE) on their estimation of the distribution parameters. We recall that the MAPE calculates the average percentage difference between the actual and predicted values of a variable, therefore providing a relative measure of error. For a sample of N units, for a generic parameter θ it is expressed through the formula:

$$\text{MAPE} = \frac{100}{N} \sum_{i=1}^N \left| \frac{\theta_i - \hat{\theta}_i}{\theta_i} \right|,$$

where $\hat{\theta}_i$ is the estimated parameter and θ_i is the true parameter. MAPE has some limitations, such as the fact that it cannot be used when actual values are zero or close to zero. This is why for the covariance matrices only the diagonal elements are considered. Results are shown in Figure 2.

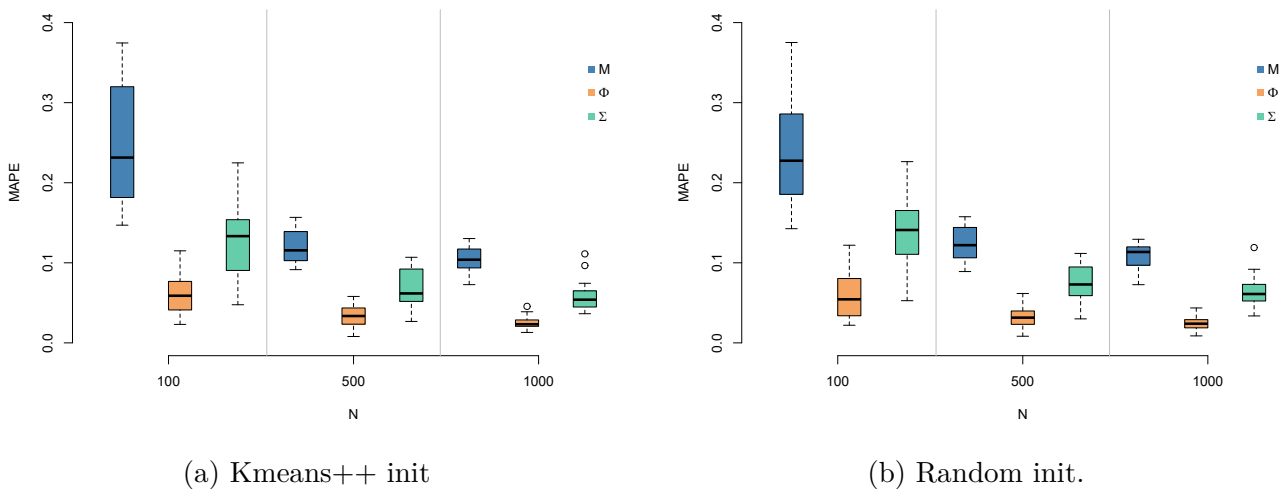


Figure 2: MAPE for increasing sample size

Regarding the parameters estimates with respect to the different initialization strategies, there is no clear difference in terms of MAPE, with the Kmeans having thinly better rendering.

Concerning overall the influence of the sample size, the model behaves as expected: as the sample size increases, the partitioning capabilities improve and tend towards the optimal error. The same happens when we observe the errors concerning the parameter

estimations for both the initialization procedures, and the median MAPE values appear to reach a stable value already for $N = 500$, while the values improve further for $N = 1000$, especially in terms of lower variability.

Overall, while the random multistart seems to provide a better partitioning, the difference is so tiny that for the rest of our experiments we will use just the Kmeans++ initialization strategy, which is less time consuming.

Last, while the general magnitude of the MAPE can seem important, it is important to recall that we use a convergence tolerance of $\varepsilon = 1 \cdot 10^{-3}$, as per Section 3.5. Better results can be found by reducing the ε , while making the execution more time-consuming.

4.4 Robustness to noise

As written in Section 4.1, we also simulated some noisy data to study the robustness of the model in presence of some noise. ARI for different noise proportions were measured and the results are visible in Figure 3. We decided to measure two quantities: the overall ARI for all the units and the ARI just for the non-noisy ones.

As we would expect, the overall quality of partitioning estimates slightly decreases as the level of noise increases, indicating that the model is actually disturbed by the noise. Interestingly but somehow to be expected, when N increases the model is more disturbed by the noise, as there are more units affected by it. Moreover, the noise affects the allocation estimation of non-noisy units as well, and again this estimation seems to be more disturbed for a larger N .

4.5 Model selection

Following the setup described in Section 4.1, by varying $N \in \{100, 500, 1000\}$ and adding increasing noise ratios $\tau \in \{0, 0.1, 0.2\}$, 9 different scenarios have derived for testing the model selection capabilities. We recall that for each scenario and each N , 20 data sets have been drawn. Model selection has been performed through BIC, as described in Section 3.6. The results are shown in Table 1.

When the sample size increases, the model converges toward the true model. However, as clearly visible in the table, the model tend to underestimate the true number of clusters when the sample size is not sufficiently large, probably due to the insufficient number of units to estimate the parameters from. Interestingly, some noise actually helps the model to recover the true model.

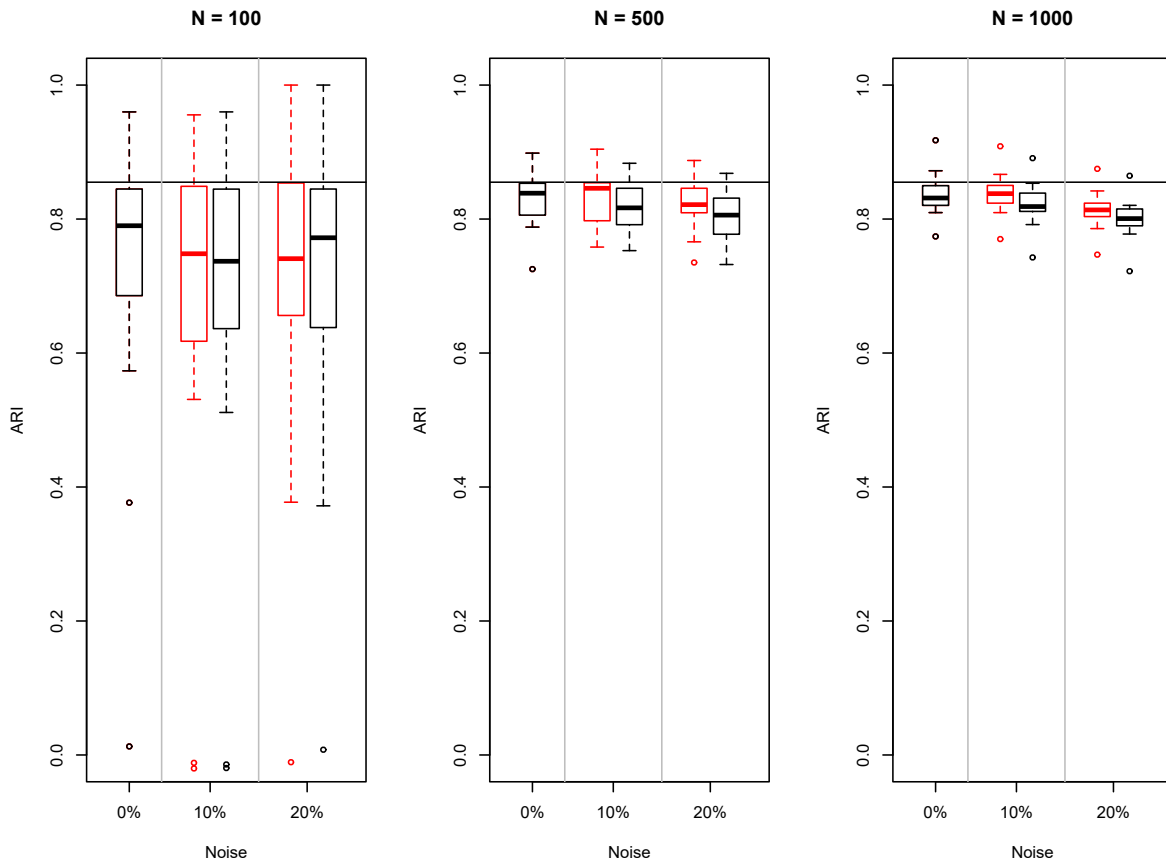


Figure 3: ARI for increasing noise proportions and increasing N .
 In red the ARI for non-noisy units, in black for all of them.

4.6 Comparison with continuous counterpart

Finally, we compared the MMM model to the classical Mixture of Matrix-Normals (MMN) model, mentioned in Section 1.1, in a version implemented by us following Viroli, 2011a. Essentially, this means treating all the different data-type equally as continuous, as it is often done by practitioners, but keeping the advantages of the matrix-variate structure. The results of the partitioning is presented in Figure 4. The hyper-parameters of the competitors have been set to be similar to the one of the MMM in terms of initialization, convergence and covariance matrix parametrization. The MMM model clearly outperforms the MMN model, independently from the sample size.

In Figure 5, we compared the MAPE values for the parameters estimation between MMM and MMN models. The difference between the two is severe, especially for the matrix of means M and the covariance matrix Σ , while moderate for Φ , due to the constraint on its determinant. The important difference of the results of the MMN model against the

N/K	Scenario $\tau = 0$				Scenario $\tau = 0.1$				Scenario $\tau = 0.2$			
	1	2	3	4	1	2	3	4	1	2	3	4
100	14	6	0	0	13	7	0	0	12	8	0	0
500	0	19	1	0	0	20	0	0	0	20	0	0
1000	0	17	3	0	0	17	2	1	0	18	2	0

Table 1: Frequency of selection of each model K by the model through BIC among the 20 simulated data sets, for increasing N . The actual value for K is 2. Kmeans++ initialization. In bold the true value for K and the most frequent K detected for each noise ratio and sample size.

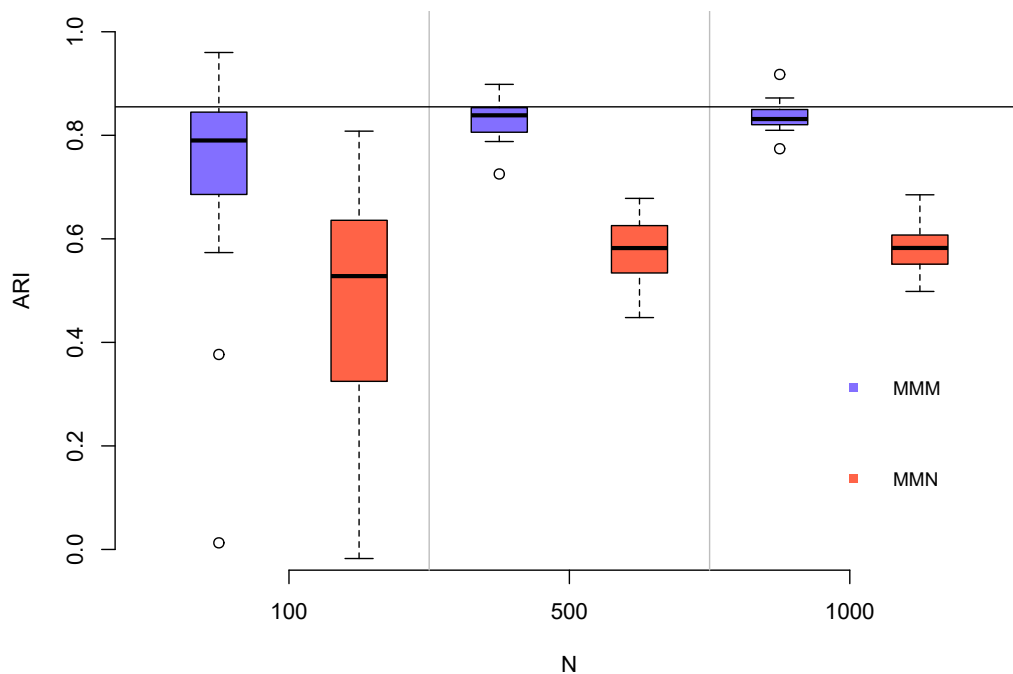
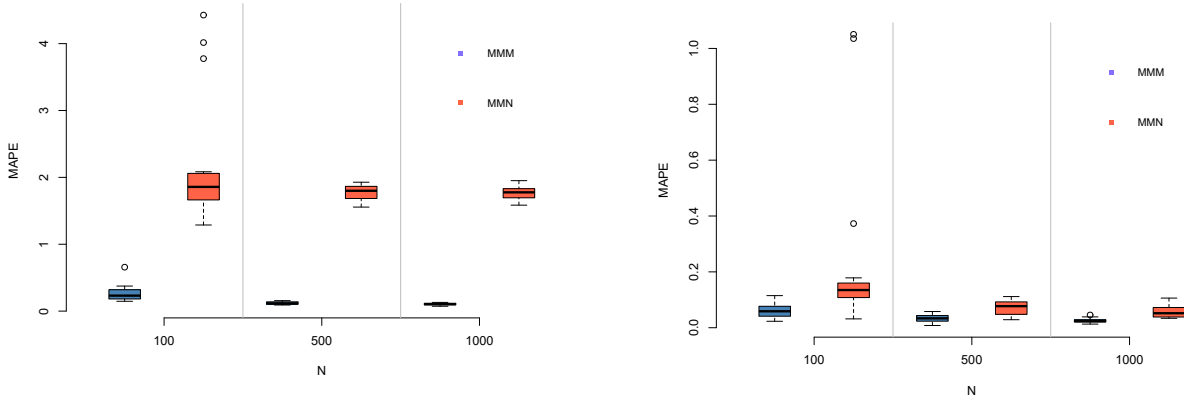


Figure 4: Comparison between the MMM and MOM models.

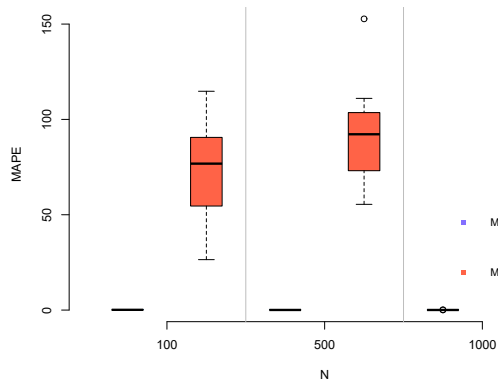
MMM one with respect to M and Σ is probably due to the count data-type variable. Indeed, without assuming the latent log-normal distribution, the values likely become too out of scale compared to the others.

Globally, the experiments described above proved that the MMM model is able to retrieve the true partitioning and to infer the true parameters, even in presence of mod-



(a) MAPE for M

(b) MAPE for Φ



(c) MAPE for Σ

Figure 5: MAPE results for parameter matrices. MMM vs MMN. Kmeans++ init. Note the difference in the scales.

erate noise. It is also able select the appropriate number of clusters through BIC when presented with enough sample units. We proved that our model outperforms its continuous matrix-variate counterpart when the latter is used to model mixed-type data, as often done by practitioners. We are now confident enough to apply the MMM model on real-world data.

5 Real-world application

5.1 Data description

The S&P500 index is a stock exchange index tracking the stock performances of 500 of the largest companies listed on stock exchange market in the United States, where each company is weighted by its market capitalization. It is one of the most commonly followed equity indices and the companies included in the index represent 80% of the total market capitalization of U.S. public companies. While investors are commonly interested in the index in its entirety, it is often the case for them to be interested in the composing companies, reputed as the best ones to invest in, in order to create specific portfolios to be used for long-term investments and wealth management.

We collected data concerning companies composing the S&P500 stock market index. Specifically, we focused on the time period going to the beginning of 2019 to the end of 2023, hence encompassing the period immediately prior and the one immediately succeeding to the COVID-19 pandemic, which went from the 30th of January 2020 to the 5th of May 2023 according to the World Health Organization (WHO) (Sarker et al., 2023). The objective of our study is to cluster companies according to their stock behavior during the pandemic period, in order to discover similar patterns during a shock period and possibly adjust our stock portfolio accordingly.

For our analysis, we collected for each year and for each listed company the following variables:

- **LogReturns:** continuous variable. The logarithm of the yearly return of the stock. The return is computed as the relative percentage change in the stock adjusted closing price between the first trading day versus the last trading day of the year. In financial analysis, log-returns are often employed instead of the simple returns as log returns have an infinite support (compared to simple returns which are lower-bounded by -100) and as they take into account the compounding effect, making them more suitable for long-term analysis.
- **Grades:** ordinal variable. The investment grade of the stock expressed by institutional investment banks. Specifically, for this study we considered the grades given by “Bank of America”, since it is the institution that releases them for most of the companies of the S&P500. The grades have three levels: “Underperform”, “Neutral” and “Buy”. Grades are given multiple times in a year and not all at the same time, so we considered their mode for each fiscal year.
- **Dividends:** binary variable. Whether the stock gave right to a dividend during the fiscal year or not, regardless of the amount .

- **Volume:** count variable. The total volume of stocks exchanged during the year. Because of the high amount of stocks that are traded during a year, we decided to count per millions of stocks exchanged. Therefore, each counted units will represent a million stocks traded. Generally, securities with higher volume are more liquid.

The data were collected making use of the Python package `pyfinance`, which downloads the data from the website “yahoo!finance”.

However, grades were not released by Bank of America for all the S&P500 companies for the entirety of the time window of our study, but just on 330 of them. We decided to reduce our survey to them. So, overall our dataset is composed of four mixed variables (continuous, ordinal, binary and count) collected for 330 observations over 5 time points (years from 2019 to 2023 included). We reorganized these data into a list of matrices.

5.2 Results

After performing our clustering algorithm with a number of clusters K ranging from 1 to 8 using Kmeans++ initialization, the model with the lowest BIC is the one with $K = 4$ (Fig. D2). The number of units in each cluster is respectively of 94, 50, 154 and 32.

The estimated parameters are reported in Table D1 for the mean M , Table D2 for the time covariances Φ and in Table D3 for the variable covariances Σ . In addition, the correlation matrices are represented by correlation plots in Figs. 8 and 9, respectively. The tickers of the companies allocated to each cluster is reported at Table D4.

In Figure 7 the evolution for the observed outcomes for each cluster is showed.

Moreover, by using the “Global Industry Classification Standard” (GICS) industrial taxonomy developed by “Standard & Poor’s” (S&P), we represented the sector composition of each cluster in Figure D1.

By performing a PCA the latent continuous embedding computed by the MMM model, we can represents the 330 units as in Figure 6a. A 3D representation is provided. For this representation, the temporal structure has been discarded and we have transformed our latent embedding for the units from 4×5 -dimensional matrices to 20-dimensional vectors. On the other hand, Figure 6b represents cluster means at each of the 5 years. Such plot allows to visualize the time evolution of each cluster.

5.3 Interpretation

First, we can get a preliminary idea by looking at Figure 6a. As we can see, of the 4 clusters, units belonging to Cluster 1 and Cluster 3 are more concentrated and closer to each other in the latent space, while the ones belonging to Cluster 2 and 4 are more spread out. This is confirmed by looking at Figure 6b, where the clusters of Cluster 1 and Cluster 3 means occupy adjacent space regions.

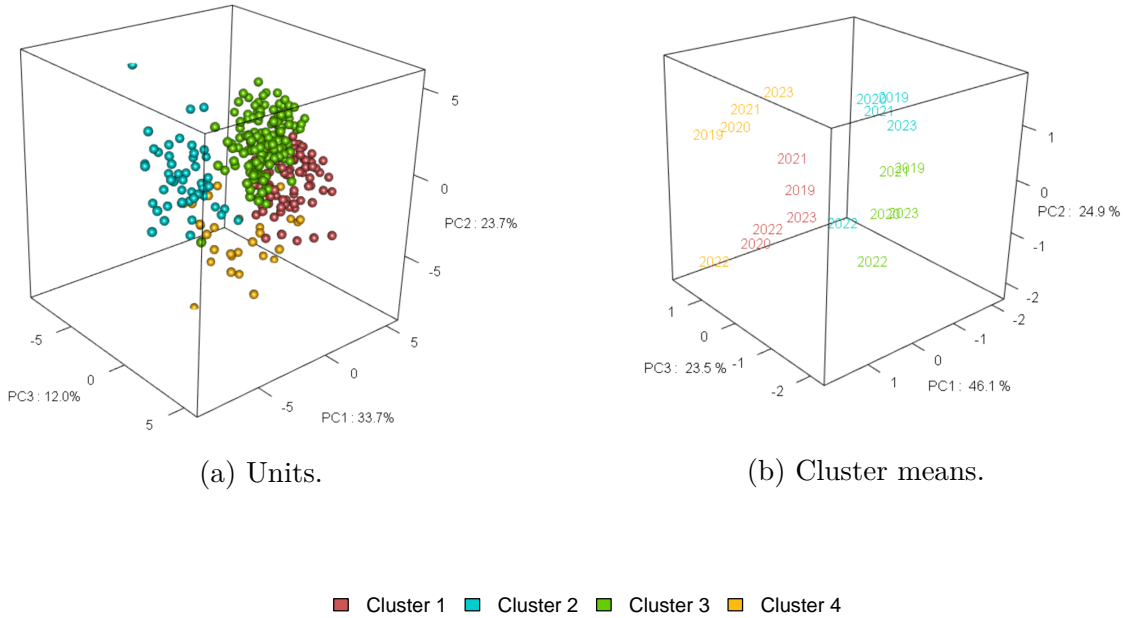


Figure 6: Units and cluster means represented through PCA.

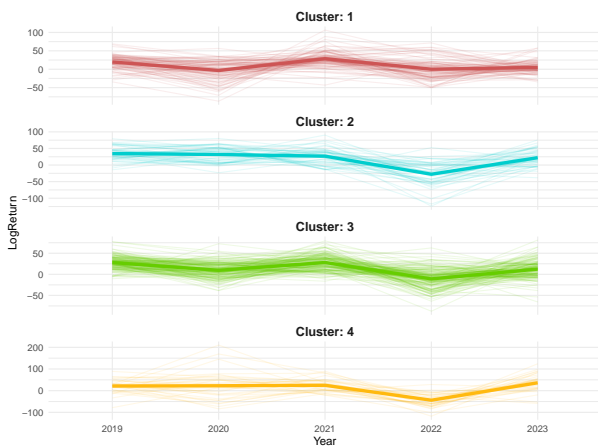
In the following, we give a summary description for Cluster 4, which we deemed be the most interesting. Interpretations for the other clusters can be found in Appendix B.

- **Cluster 4:** 32 units.

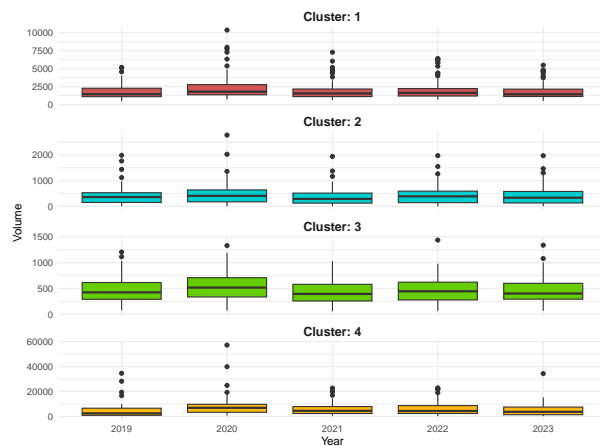
- **Means:** the cluster is qualified by generally constant strong values for LogReturn, with the exception of 2022, where the cluster has the lowest negative value. The cluster also has the second highest values for Grade and the highest values for Volume. The values for Dividend are small and fluctuate around zero in time suggesting heterogeneity in the cluster regarding this variable.
- **Correlation in time:** the cluster is characterized the second strongest correlations overall.
- **Correlation among variables:** the main feature of the cluster concerning variables correlation is the absence of a negative correlation between volume and LogReturn, while a weak negative correlation between Dividend and LogReturn is estimated.

Cluster 4 is defined by its high value of the variable Volume compared to the others. The values of LogReturn are more stable in time, except for 2022. The value of Dividens float around zero, and Figure 7 shows us that the dividend distribution is almost evenly split for most of the years.

It is also the only cluster to have a negative correlation between Dividend and LogReturn, implying that stocks with higher returns are also the ones with no dividends. This paradox can be explained by looking at the sector distribution in Figure 7 : a majority of the companies whose stocks are allocated to Cluster 4 belong to sectors such as “Technology” and “Consumer Cyclical”, and when we look at Table D4 we realize it includes companies like Amazon, Tesla, Netflix, Nvidia, AMD and Moderna, that do not allocate dividends but prefer to reinvest their profit in R&D.



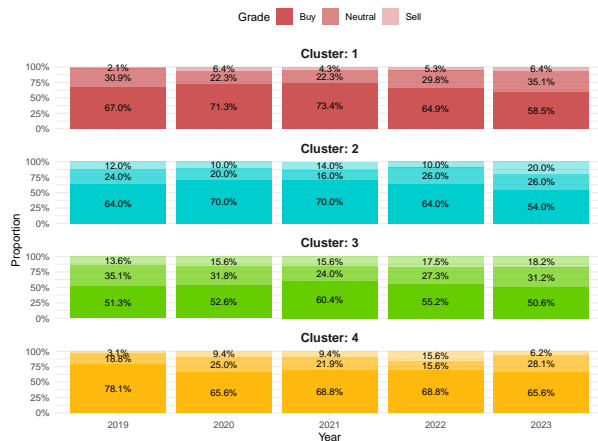
(a) LogReturns



(b) Volume



(c) Dividends



(d) Grades

Figure 7: Observed variables values for each cluster. Note that for graphical reason in plots (a) and (b) the company NVIDIA has been removed from the set, due to its out-of-scale values compared to the others companies.

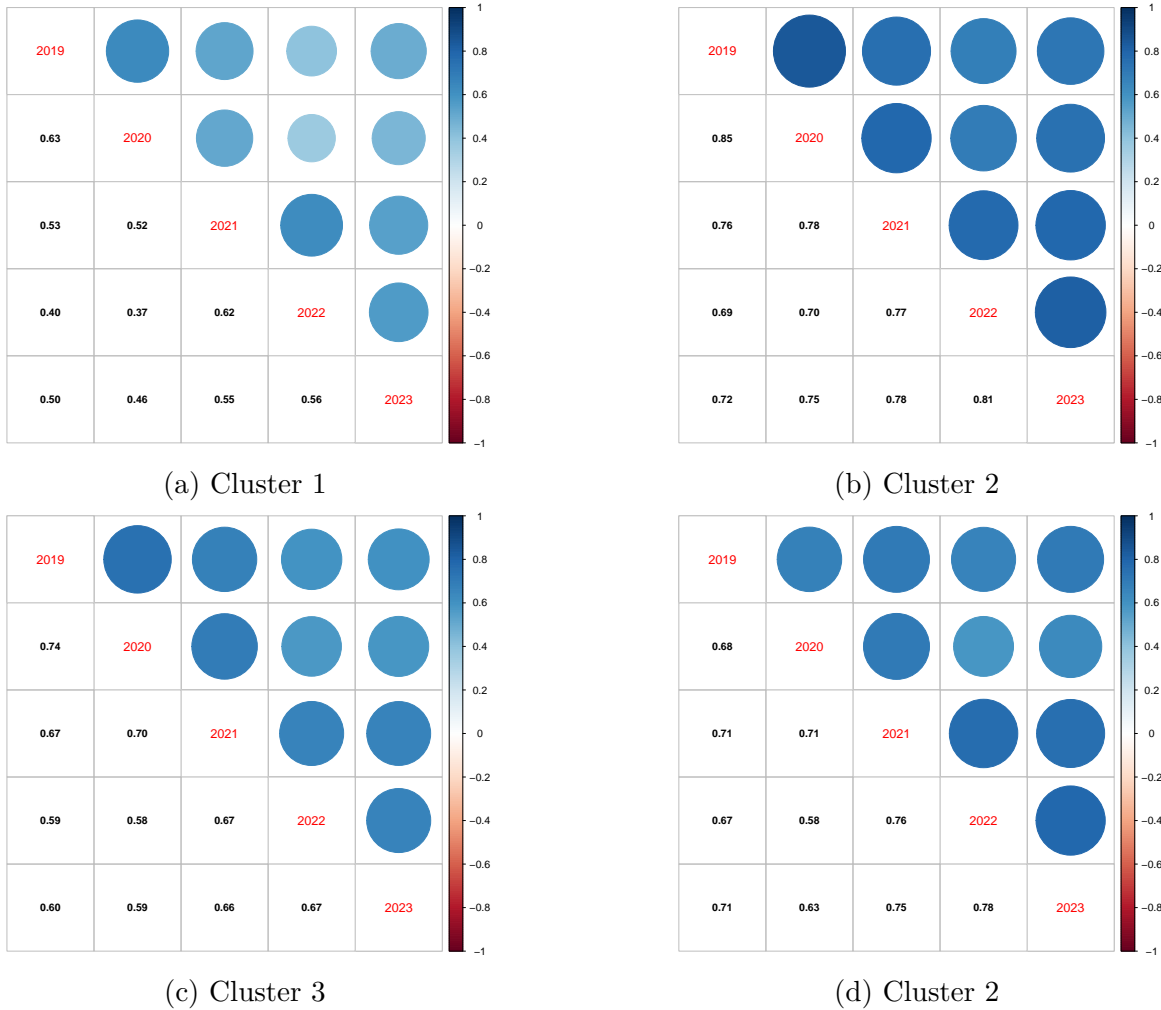


Figure 8: Clusters' corr-plots among years.

6 Conclusions

In this work we have presented a novel approach for modeling longitudinal mixed-type data with unobserved heterogeneity. The model presented does not require the conditional independence assumption. The matrix-variate structure allows for a more parsimonious modeling of multivariate longitudinal data than other models in the literature. Also, it can explicitly model the temporal structure and the association among the responses, that can vary among clusters. An MCMC-EM algorithm to perform inference has been proposed and described. The efficacy of the algorithm has been tested on synthetic data under different sample sizes and different noise ratios. We proved the goodness of this framework to cluster longitudinal mixed-type data and to get clusters that are easy to interpret and to work with even by non-statisticians in a real-world example.

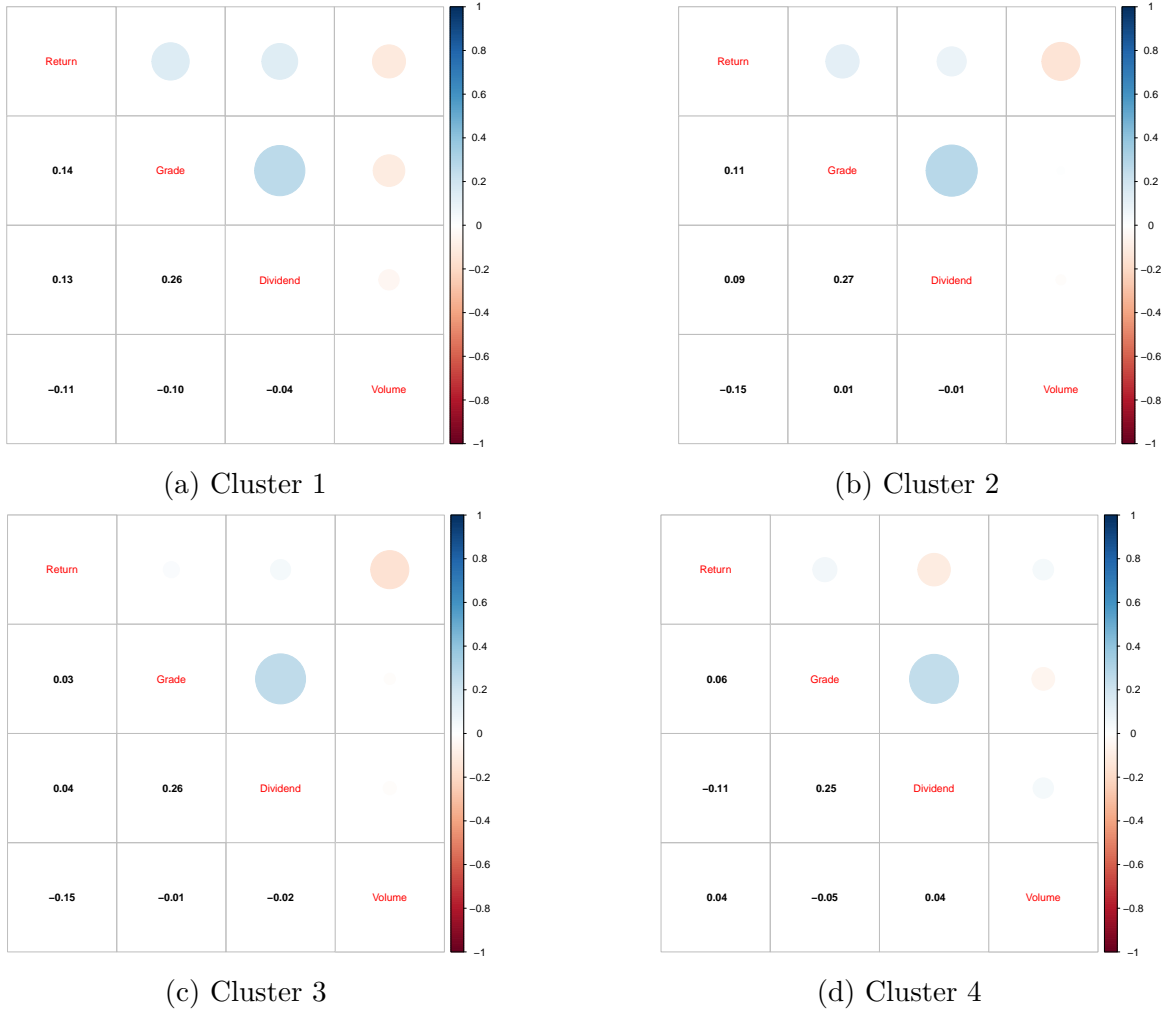


Figure 9: Clusters' corr-plots among variables.

However, the proposed model has some limitations. In this paper we focused only on the simplest structure of matrix-normal distribution. While considerably more parsimonious than a mixture of multivariate normal distributions, the model seems sensitive to small sample sizes, since, as the number of clusters increases, the number of parameters to estimate can still become troublesome. To improve this aspect, the covariance matrices can be further decomposed to obtain more flexible and parsimonious models, as done for example in [Anderlucci et al., 2015](#) and in [Sarkar et al., 2020](#). Another solution to this problem can be the one proposed by [Cappozzo et al., 2023](#).

Similarly, the matrix-variate structure is not just inherent to multivariate longitudinal data, but can actually be found in many other applications. The MMM model can be

employed in such cases as well, with minimal adjustments required. Moreover, EM algorithm can be leveraged to extend the model to deal with incomplete data under the missing at random (MAR).

Finally, one could as well think of employing, with proper adjustments, different underlying continuous distributions, such as heavy-tailed (Tomarchio et al., 2020), skewed (Gallaughner et al., 2018, Melnykov et al., 2018) or t-student (Doğru et al., 2016) distributions to endow the clustering model with different desired properties.

Acknowledgment

This work has been realised thanks to the financial support provided by Project IADoc@UdL of the University of Lyon and Université Lumière - Lyon 2 as part of the call for “doctoral contracts in artificial intelligence 2020” (ANR-20-THIA-0007-01).

References

- [1] William M. Rand. “Objective Criteria for the Evaluation of Clustering Methods”. In: *Journal of the American Statistical Association* 66.336 (Dec. 1971), pp. 846–850. ISSN: 0162-1459. DOI: [10.1080/01621459.1971.10482356](https://doi.org/10.1080/01621459.1971.10482356).
- [2] A. P. Dempster et al. “Maximum Likelihood from Incomplete Data Via the EM Algorithm”. In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (Sept. 1977), pp. 1–22. ISSN: 0035-9246. DOI: [10.1111/j.2517-6161.1977.tb01600.x](https://doi.org/10.1111/j.2517-6161.1977.tb01600.x).
- [3] Gideon Schwarz. “Estimating the Dimension of a Model”. In: *Annals of Statistics* 6.2 (Mar. 1978), pp. 461–464. ISSN: 0090-5364. DOI: [10.1214/aos/1176344136](https://doi.org/10.1214/aos/1176344136).
- [4] B.S. Everitt. *Introduction to Latent Variable Models*. Chapman and Hall, 1984.
- [5] Kaye E. Basford et al. “The mixture method of clustering applied to three-way data”. In: *Journal of Classification* 2.1 (Dec. 1985), pp. 109–125. ISSN: 1432-1343. DOI: [10.1007/BF01908066](https://doi.org/10.1007/BF01908066).
- [6] Arjun Kumar Gupta et al. *Matrix Variate Distributions*. Chapman and Hall/CRC, 2000.
- [7] David Arthur et al. “k-means++: the advantages of careful seeding”. In: *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. USA: Society for Industrial and Applied Mathematics, Jan. 2007, pp. 1027–1035. ISBN: 978-0-89871624-5. DOI: [10.5555/1283383.1283494](https://doi.org/10.5555/1283383.1283494).
- [8] Rolando De la Cruz-Mesia et al. “Model-based clustering for longitudinal data”. In: *Computational Statistics & Data Analysis* 52.3 (2008), pp. 1441–1457.

- [9] Geoffrey J McLachlan et al. *The EM algorithm and extensions*. John Wiley & Sons, 2008.
- [10] Nilam Ram et al. “Methods and measures: Growth mixture modeling: A method for identifying differences in longitudinal change among unobserved groups”. In: *International journal of behavioral development* 33.6 (2009), pp. 565–576.
- [11] Paul D. McNicholas et al. “Model-based clustering of longitudinal data”. In: *Canadian Journal of Statistics / La Revue Canadienne de Statistique* 38.1 (Mar. 2010), pp. 153–168. ISSN: 0319-5724. DOI: [10.1002/cjs.10047](https://doi.org/10.1002/cjs.10047).
- [12] Lynette Hunt et al. “Clustering mixed data”. In: *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 1.4 (2011), pp. 352–361.
- [13] Cinzia Viroli. “Finite mixtures of matrix normal distributions for classifying three-way data”. In: *Statistics and Computing* 21.4 (Oct. 2011), pp. 511–522. ISSN: 1573-1375. DOI: [10.1007/s11222-010-9188-x](https://doi.org/10.1007/s11222-010-9188-x).
- [14] Cinzia Viroli. “Model based clustering for three-way data structures”. In: *Bayesian Analysis* 6.4 (Dec. 2011), pp. 573–602. ISSN: 1936-0975. DOI: [10.1214/11-BA622](https://doi.org/10.1214/11-BA622).
- [15] Ahmed M. Gad et al. “Generalized Linear Mixed Models for Longitudinal Data”. In: *International Journal of Probability and Statistics* 1.3 (2012), pp. 41–47. DOI: [10.5923/j.ijps.20120103.03](https://doi.org/10.5923/j.ijps.20120103.03).
- [16] Arnost Komarek et al. “Clustering for multivariate continuous and discrete longitudinal data”. In: *The Annals of Applied Statistics* (2013), pp. 177–200.
- [17] Ccile Proust-Lima et al. “Analysis of multivariate mixed longitudinal data: a flexible latent process approach”. In: *British Journal of Mathematical and Statistical Psychology* 66.3 (2013), pp. 470–487.
- [18] Daniel Manrique-Vallier. “Longitudinal mixed membership trajectory models for disability survey data”. In: *The Annals of Applied Statistics* 8.4 (2014), p. 2268.
- [19] Laura Anderlucci et al. “Covariance pattern mixture models for the analysis of multivariate heterogeneous longitudinal data”. In: *Annals of Applied Statistics* 9.2 (June 2015), pp. 777–800. ISSN: 1932-6157. DOI: [10.1214/15-AOAS816](https://doi.org/10.1214/15-AOAS816).
- [20] Liesbeth Bruckers et al. “A clustering algorithm for multivariate longitudinal data”. In: *Journal of Biopharmaceutical Statistics* (July 2016). URL: <https://www.tandfonline.com/doi/full/10.1080/10543406.2015.1052476?src=recsys>.
- [21] Fatma Zehra Dođru et al. “Finite mixtures of matrix variate t distributions”. In: *Gazi University Journal of Science* 29.2 (2016), pp. 335–341.
- [22] Damien McParland et al. “Model based clustering for mixed data: clustMD”. In: *Advances in Data Analysis and Classification* 10.2 (June 2016), pp. 155–169. ISSN: 1862-5355. DOI: [10.1007/s11634-016-0238-x](https://doi.org/10.1007/s11634-016-0238-x).

- [23] Jie Cheng et al. “High-Dimensional Mixed Graphical Models”. In: *Journal of Computational and Graphical Statistics* (Apr. 2017). DOI: [10.1080/10618600.2016.1237362](https://doi.org/10.1080/10618600.2016.1237362).
- [24] Matthieu Marbac et al. “Model-based clustering of Gaussian copulas for mixed data”. In: *Communications in Statistics-Theory and Methods* 46.23 (2017).
- [25] Silvia Cagnone et al. “Multivariate Latent Variable Transition Models of Longitudinal Mixed Data: An Analysis on Alcohol Use Disorder”. In: *Journal of the Royal Statistical Society Series C: Applied Statistics* 67.5 (Nov. 2018), pp. 1399–1418. ISSN: 0035-9254. DOI: [10.1111/rssc.12285](https://doi.org/10.1111/rssc.12285).
- [26] Michael P. B. Gallagher et al. “Finite mixtures of skewed matrix variate distributions”. In: *Pattern Recognition* 80 (Aug. 2018), pp. 83–93. ISSN: 0031-3203. DOI: [10.1016/j.patcog.2018.02.025](https://doi.org/10.1016/j.patcog.2018.02.025).
- [27] Volodymyr Melnykov et al. “On model-based clustering of skewed matrix data”. In: *Journal of Multivariate Analysis* 167 (Sept. 2018), pp. 181–194. ISSN: 0047-259X. DOI: [10.1016/j.jmva.2018.04.007](https://doi.org/10.1016/j.jmva.2018.04.007).
- [28] Amir Ahmad et al. “Survey of State-of-the-Art Mixed Data Clustering Algorithms”. In: *IEEE Access* 7 (Mar. 2019), pp. 31883–31902. ISSN: 2169-3536. DOI: [10.1109/ACCESS.2019.2903568](https://doi.org/10.1109/ACCESS.2019.2903568).
- [29] Francesco Bartolucci et al. *Latent Markov models for longitudinal data*. Chapman and Hall/CRC, 2019.
- [30] Charles Bouveyron et al. *Model-Based Clustering and Classification for Data Science: With Applications in R*. Cambridge, England, UK: Cambridge University Press, June 2019. ISBN: 978-1-10864418-1. DOI: [10.1017/9781108644181](https://doi.org/10.1017/9781108644181).
- [31] Volodymyr Melnykov et al. “Studying crime trends in the USA over the years 2000–2012”. In: *Advances in Data Analysis and Classification* 13.1 (Mar. 2019), pp. 325–341. ISSN: 1862-5355. DOI: [10.1007/s11634-018-0326-1](https://doi.org/10.1007/s11634-018-0326-1).
- [32] Margot Selosse, Julien Jacques, Christophe Biernacki, and Florence Cousson-Glie. “Analysing a quality-of-life survey by using a co-clustering model for ordinal data and some dynamic implications”. In: *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 68.5 (Nov. 2019), pp. 1327–1349. ISSN: 0035-9254. DOI: [10.1111/rssc.12365](https://doi.org/10.1111/rssc.12365).
- [33] Marco Corneli et al. “Co-Clustering of Ordinal Data via Latent Continuous Random Variables and Not Missing at Random Entries”. In: *Journal of Computational and Graphical Statistics* 29.4 (Oct. 2020), pp. 771–785. ISSN: 1061-8600. DOI: [10.1080/10618600.2020.1739533](https://doi.org/10.1080/10618600.2020.1739533).
- [34] Shuchismita Sarkar et al. “On parsimonious models for modeling matrix data”. In: *Computational Statistics & Data Analysis* 142 (Feb. 2020), p. 106822. ISSN: 0167-9473. DOI: [10.1016/j.csda.2019.106822](https://doi.org/10.1016/j.csda.2019.106822).

- [35] Margot Selosse, Julien Jacques, and Christophe Biernacki. “Model-based co-clustering for mixed type data”. In: *Computational Statistics & Data Analysis* 144 (Apr. 2020). ISSN: 0167-9473. DOI: [10.1016/j.csda.2019.106866](https://doi.org/10.1016/j.csda.2019.106866).
- [36] Salvatore D. Tomarchio et al. “Two new matrix-variate distributions with application in model-based clustering”. In: *Computational Statistics & Data Analysis* 152 (Dec. 2020), p. 107050. ISSN: 0167-9473. DOI: [10.1016/j.csda.2020.107050](https://doi.org/10.1016/j.csda.2020.107050).
- [37] Yang Wang et al. “On variable selection in matrix mixture modelling”. In: *Stat* 9.1 (Jan. 2020), e278. ISSN: 2049-1573. DOI: [10.1002/sta4.278](https://doi.org/10.1002/sta4.278).
- [38] Jan Vávra et al. “Clustering based on multivariate mixed type longitudinal data with an application to the EU-SILC database”. In: *22nd European Young Statisticians Meeting*. 2021, p. 148.
- [39] Xuwen Zhu et al. “MatTransMix: an R Package for Matrix Model-Based Clustering and Parsimonious Mixture Modeling”. In: *Journal of Classification* 39.1 (Mar. 2022), pp. 147–170. ISSN: 1432-1343. DOI: [10.1007/s00357-021-09401-9](https://doi.org/10.1007/s00357-021-09401-9).
- [40] Leonardo Salvatore Alaimo et al. “A Comparison of Migrant Integration Policies via Mixture of Matrix-Normals”. In: *Social Indicators Research* 165.2 (Jan. 2023), pp. 473–494. ISSN: 1573-0921. DOI: [10.1007/s11205-022-03024-2](https://doi.org/10.1007/s11205-022-03024-2).
- [41] Andrea Cappozzo et al. “Sparse model-based clustering of three-way data via lasso-type penalties”. In: *ArXiv e-prints* (July 2023). DOI: [10.48550/arXiv.2307.10673](https://doi.org/10.48550/arXiv.2307.10673). eprint: [2307.10673](https://arxiv.org/abs/2307.10673).
- [42] Young-Geun Choi et al. “Model-Based Clustering of Mixed Data With Sparse Dependence”. In: *IEEE Access* 11 (July 2023), pp. 75945–75954. DOI: [10.1109/ACCESS.2023.3296790](https://doi.org/10.1109/ACCESS.2023.3296790).
- [43] Rapyt Sarker et al. “The WHO declares COVID-19 is no longer a public health emergency of international concern: benefits, challenges, and necessary precautions to come back to normal life”. In: *International Journal of Surgery* 109.9 (May 2023), p. 2851. DOI: [10.1097/JS9.0000000000000513](https://doi.org/10.1097/JS9.0000000000000513).
- [44] Anjali Silva et al. “Finite Mixtures of Matrix Variate Poisson-Log Normal Distributions for Three-Way Count Data”. In: *Bioinformatics* (Apr. 2023), btad167. ISSN: 1367-4811. DOI: [10.1093/bioinformatics/btad167](https://doi.org/10.1093/bioinformatics/btad167).
- [45] Junyi Zhou et al. “clusterMLD: An Efficient Hierarchical Clustering Method for Multivariate Longitudinal Data”. In: *Journal of Computational and Graphical Statistics* (July 2023). ISSN: 1061-8600. DOI: [10.1080/10618600.2022.2149540](https://doi.org/10.1080/10618600.2022.2149540).
- [46] Francesco Amato et al. “Clustering longitudinal ordinal data via finite mixture of matrix-variate distributions”. In: *Statistics and Computing* 34.2 (Apr. 2024). ISSN: 1573-1375. DOI: [10.1007/s11222-024-10390-z](https://doi.org/10.1007/s11222-024-10390-z).

- [47] Sjoerd Hermes et al. “Copula Graphical Models for Heterogeneous Mixed Data”. In: *Journal of Computational and Graphical Statistics* (Jan. 2024). ISSN: 1061-8600. DOI: [10.1080/10618600.2023.2289545](https://doi.org/10.1080/10618600.2023.2289545).
- [48] Francis K. C. Hui et al. “Simultaneous Coefficient Clustering and Sparsity for Multivariate Mixed Models”. In: *Journal of Computational and Graphical Statistics* (Oct. 2024). ISSN: 1061-8600. DOI: [10.1080/10618600.2024.2402904](https://doi.org/10.1080/10618600.2024.2402904).
- [49] William Ruth. *A review of Monte Carlo-based versions of the EM algorithm*. 2024. arXiv: [2401.00945](https://arxiv.org/abs/2401.00945) [stat.CO]. URL: <https://arxiv.org/abs/2401.00945>.
- [50] Stan Development Team. *RStan: the R interface to Stan*. R package version 2.32.6. 2024. URL: <https://mc-stan.org/>.
- [51] Jan Vvra et al. “Clusterwise multivariate regression of mixed-type panel data”. In: *Statistics and Computing* 34.1 (Feb. 2024), pp. 1–20. ISSN: 1573-1375. DOI: [10.1007/s11222-023-10304-5](https://doi.org/10.1007/s11222-023-10304-5).

Appendices

A E-step computations

Here we will expand the computations presented in Section 3.1.

For Equation 10, the matrix-variate expectation related to count data can be computed by defining $z_i^\gamma \in \mathbb{R}^{GT \times 1}$ as the vectorized version of Z_i^γ and computing

$$\begin{aligned} \hat{m}_{ik}^{\gamma, (s+1)} &:= \mathbb{E}(z_i^\gamma | \ell_{ik} = 1, \mathbf{Y}, \hat{\Theta}^{(s)}) = \\ &= \int_{\mathbb{R}} z_i^\gamma \cdot \frac{\prod_t^T \prod_g^G \mathcal{P}(y_{igt}^\gamma | \exp(z_{igt}^\gamma)) \cdot \mathcal{MN}_{GT}(z_i^\gamma | \text{vec}(M_k^{(s), \gamma | \alpha, \beta}), \Sigma_k^{(s), \gamma | \alpha, \beta} \otimes \Phi_k^{(s)})}{\int_{\mathbb{R}} \prod_t^T \prod_g^G \mathcal{P}(y_{igt}^\gamma | \exp(z_{igt}^\gamma)) \cdot \mathcal{MN}_{GT}(z_i^\gamma | \text{vec}(M_k^{(s), \gamma | \alpha, \beta}), \Sigma_k^{(s), \gamma | \alpha, \beta} \otimes \Phi_k^{(s)}) dz_i^\gamma} dz_i^\gamma. \end{aligned} \quad (16)$$

This integral does not have any close form solution, so we resort to numerically compute it through the No-U-Turn sampler implemented in the R package `Rstan`.

Then, $\hat{M}_{ik}^{\gamma, (s+1)} := \text{vec}_{G \times T}^{-1}(\hat{m}_{ik}^{\gamma, (s+1)})$, $\text{vec}_{G \times T}^{-1}$ being the inverse of the vectorization function, i.e. the function mapping from a GT -dimensional vector to a $O \times T$ matrix.

The matrix-variate expectation related to categorical data can be computed by defining $z_i^\beta \in \mathbb{R}^{OT \times 1}$ as the vectorized version of Z_i^β and computing

$$\hat{m}_{ik}^{\beta, (s+1)} := \mathbb{E}(z_i^\beta | \ell_{ik} = 1, \mathbf{Y}, \hat{\Theta}^{(s)}) = \int_{\Omega_r} z_i^\beta \mathcal{MN}_{OT}(z_i^\beta | \text{vec}(M_k^{(s), \beta | \alpha}), \Sigma_k^{(s), \beta | \alpha} \otimes \Phi_k^{(s)}) dz_i^\beta \quad (17)$$

through the use of a Gibbs sampler to sample from a truncated multivariate normal distribution.

Then, as we did for count data; we map the estimated values back to a matrix form as $\hat{M}_{ik}^{\beta, (s+1)} := \text{vec}_{O \times T}^{-1}(\hat{m}_{ik}^{\beta, (s+1)})$.

For Equation 11, to compute $D_{ik}^{(s)}$, we start by defining $\hat{\varphi}_{k, gd}^{(s)}$ as the $(g, d)^{th}$ element of $\hat{\Phi}_k^{-1(s)}$. Then, the $(h, t)^{th}$ element of $Z_i^\beta \Phi_k^{-1} Z_i^{\beta \top}$ would be $\sum_{d=1}^T \sum_{g=1}^T z_{i, hg}^\beta \hat{\varphi}_{k, gd}^{(s)} z_{i, td}^\beta$ and we would get

$$\begin{aligned} \hat{D}_{ik}^{(s)} &:= \mathbb{E}(Z_i^\beta \Phi_k^{-1} Z_i^{\beta \top} | \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y}) = \\ &= \left(\sum_{d=1}^T \sum_{g=1}^T \hat{S}_{ik, [(g-1)O+h, (d-1)O+t]}^{\beta, (s+1)} \hat{\varphi}_{k, gd}^{(s)} \right)_{h,t}, \end{aligned} \quad (18)$$

where we make use of the the elements of

$$\hat{S}_{ik}^{\beta,(s+1)} := \mathbb{E}(z_i^\beta z_i^{\beta\top} | \ell_{ik} = 1, \mathbf{Y}, \hat{\Theta}^{(s)}) = \int_{\Omega_r} z_i^\beta z_i^{\beta\top} \mathcal{M}\mathcal{N}_{OT}(z_i^\beta | \text{vec}(M_k^{(s),\beta|\alpha}), \Sigma_k^{(s),\beta|\alpha} \otimes \Phi_k^{(s)}) dz_i^\beta. \quad (19)$$

The samples generated to calculate the first moment $m_{ik}^{\beta,(s+1)}$ can be reused to compute the matrix $\hat{S}_{ik}^{\beta,(s+1)}$, that can be approximated by calculating the inner product of the vectors used to compute $m_{ik}^{\beta,(s+1)}$ then calculating the sample mean of these inner products.

Similarly, for $\hat{B}_{ik}^{(s)}$ the $(h, t)^{th}$ element of $Z_i^\gamma \Phi_k^{-1} Z_i^{\gamma\top}$ would be $\sum_{d=1}^T \sum_{g=1}^T z_{i,hg}^\gamma \varphi_{k,gd} z_{i,td}^\gamma$ and we would get

$$\begin{aligned} \hat{B}_{ik}^{(s)} &:= \mathbb{E}(Z_i^\gamma \Phi_k^{-1} Z_i^{\gamma\top} | \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y}) = \\ &= \left(\sum_{d=1}^T \sum_{g=1}^T \hat{S}_{ik,[(g-1)G+h,(d-1)G+t]}^{\gamma,(s+1)} \hat{\varphi}_{k,gd}^{(s)} \right)_{h,t}, \end{aligned} \quad (20)$$

where we make use of the the elements of

$$\begin{aligned} \hat{S}_{ik}^{\gamma,(s+1)} &:= \mathbb{E}(z_i^\gamma z_i^{\gamma\top} | \ell_{ik} = 1, \mathbf{Y}, \hat{\Theta}^{(s)}) = \\ &= \int_{\mathbb{R}} z_i^\gamma z_i^{\gamma\top} \cdot \frac{\prod_t \prod_g \mathcal{P}(y_{igt}^\gamma | \exp(z_{igt}^\gamma)) \cdot \mathcal{M}\mathcal{N}_{GT}(z_i^\gamma | \text{vec}(M_k^{(s),\gamma|\alpha,\beta}), \Sigma_k^{(s),\gamma|\alpha,\beta} \otimes \Phi_k^{(s)})}{\int_{\mathbb{R}} \prod_t \prod_g \mathcal{P}(y_{igt}^\gamma | \exp(z_{igt}^\gamma)) \cdot \mathcal{M}\mathcal{N}_{GT}(z_i^\gamma | \text{vec}(M_k^{(s),\gamma|\alpha,\beta}), \Sigma_k^{(s),\gamma|\alpha,\beta} \otimes \Phi_k^{(s)}) dz_i^\gamma} dz_i^\gamma. \end{aligned} \quad (21)$$

As before, the samples generated to calculate the first moment $\hat{m}_{ik}^{\gamma,(s+1)}$ can be reused to compute the matrix $\hat{S}_{ik}^{\gamma,(s+1)}$ by calculating the mean of the inner product between them.

Finally, for Equation 12, let us define by $\hat{\sigma}_{k,gd}^{(s),\beta\beta}$ the $(g, d)^{th}$ element of the block $\hat{\Sigma}_k^{-1(s),\beta\beta}$. Then, the $(h, t)^{th}$ element of $Z_i^{\beta\top} \Sigma_k^{\beta\beta} Z_i^\beta$ is $\sum_{d=1}^O \sum_{g=1}^O z_{i,gh} \hat{\sigma}_{k,gd}^{(s),\beta\beta} z_{i,dt}$, and we get

$$\begin{aligned} \hat{C}_{ik}^{(s)} &:= \mathbb{E}(Z_i^{\beta\top} \Sigma_k^{\beta\beta} Z_i^\beta | \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y}) = \\ &= \left(\sum_{d=1}^O \sum_{g=1}^O \hat{S}_{ik,[(h-1)J+g,(t-1)J+d]}^{\beta,(s+1)} \hat{\sigma}_{k,gd}^{(s),\beta\beta} \right)_{h,t}. \end{aligned} \quad (22)$$

For $\hat{A}_{ik}^{(s)}$, let indicate by $\hat{\sigma}_{k,gd}^{(s),\gamma\gamma}$ the $(g, d)^{th}$ element of block $\hat{\Sigma}_k^{-1(s),\gamma\gamma}$. Then, the $(h, t)^{th}$ element of $Z_i^{\gamma\top} \Sigma_k^{\gamma\gamma} Z_i^\gamma$ is $\sum_{d=1}^O \sum_{g=1}^O z_{i,gh} \hat{\sigma}_{k,gd}^{(s),\gamma\gamma} z_{i,dt}$, and we get

$$\begin{aligned}
\hat{A}_{ik}^{(s)} &:= \mathbb{E}(Z_i^{\gamma\top} \Sigma_k^{\gamma\gamma} Z_i^{\gamma} | \ell_{ik} = 1, \hat{\Theta}^{(s)}, \mathbf{Y}) = \\
&= \left(\sum_{d=1}^O \sum_{g=1}^O \hat{S}_{ik, [(h-1)J+g, (t-1)J+d]}^{\gamma, (s+1)} \hat{\sigma}_{k, gd}^{(s), \gamma\gamma} \right)_{h,t}. \tag{23}
\end{aligned}$$

B Cluster interpretation

In this section interpretations for other clusters referred in Section 5.3 are given.

- **Cluster 1:** 94 units.
 - **Means:** the cluster means show the highest values for Dividend and Grade, and the second highest for Volume. The results for LogReturn are more shaded: the cluster has the lowest mean for 2019, 2020 (the only one negative for that year) and 2023. At the same time, it is the only cluster to have non-negative LogReturns for 2022.
 - **Correlation in time:** the cluster is characterized by a fading and weaker correlations among times than other clusters, especially regarding 2022 to the previous years.
 - **Correlation among variables:** the cluster is characterized by feeble correlations among Returns, Grade and Dividend, yet these correlations are stronger than in other clusters. Some soft negative correlations are estimated between Volume, Grade and Return.

We can describe Cluster 1 as the cluster of more “traditional” stocks. Stocks belonging to this cluster have good grades, usually grant dividends and are among the most exchanged, ensuring good liquidity.

By looking at Figure D1a, we can notice that the cluster is the ones with more variety of composing sectors. This might explain why it is the only cluster that experienced a fall in LogReturns in 2020, at the height of the COVID-19 pandemic and of the consequent lock-downs, which had a major impact on more traditional sectors. Figure 7 suggests that indeed the stocks gave right to dividends even for the entirety of them in 2019 and 2020. We can also point out to the fact that during 2020 and 2021 the percentage of stocks marked as “Buy” for this cluster increased, probably in view of the end of toughest pandemic period and in light of the lower prices of the stocks. The grades distribution changes during 2022 and 2023 mostly in favor of “Neutral”. The correlations among times suggest that the behaviour is less constant in time with respect of the other clusters. Moreover, the negative correlations between Volume, LogReturns and Grade may indicate that the increase in

volume exchange is generally related to selling, as the volume increase when grades and returns decreases.

- **Cluster 2:** 50 units.
 - **Means:** the cluster has the highest means regarding LogReturns for the first two years and the second most important negative value for 2022. It is the only cluster with relatively strong negative values for Dividend. It has also the lowest values for Volume.
 - **Correlation in time:** it is the cluster with the strongest positive correlation in time.
 - **Correlation variables:** the cluster is characterized by the presence of weak correlation between LogReturn, Grade and Dividend, and of a weak negative correlation between Volume and LogReturn.

Cluster 2 has the main characteristics to be the only cluster with negative values for Dividend. A look at Figure 7 shows us that indeed that almost none of the stocks allocated to the cluster gave right to a dividend, a situation that slightly improves in 2023. The low values for Volume compared to the other clusters indicate that the stocks in this cluster are among the less exchanged. The grades distribution show that there is a high percentage of stocks marked as “Buy” until 2021, but it decreases and in 2023 the cluster has the highest percentage of stocks marked as “Underperform”. 2022 appears to be a bad year for the stocks belonging to the cluster, but with the expect of this year the cluster has the most stable values for LogReturn. The sector composition of the cluster shows a dominance of the sectors “Healthcare” and “Technology”, which might explain the good performance during the pandemic, as these sectors were among the ones to actually profit during the pandemic. The same reason might explain the 2022 performance, where staff lay-offs and decrease in investments due to over-investments during the pandemics hit particularly the IT sector.

- **Cluster 3:** 154 units.
 - **Means:** the cluster has the second highest means for LogReturns for 2019 and 2021, and the lowest negative value for 2022. It has the second highest values for Dividend and the second smallest values for Volume.
 - **Correlation in time:** the cluster has the overall strong positive correlations in time.
 - **Correlation among variables:** the cluster is mainly characterized by the weak negative correlation between Volume and LogReturn, and the absence of other meaningful correlations.

Cluster 3 can be seen as cluster between Cluster 1 and Cluster 2: both Volume and Grade have values in between the two, and the same can be almost be said for LogReturn. The main exception to this description is Dividend, since for Cluster 3 the values are high, and if we look at Figure 7 almost 100% of the stocks gave right to a dividend. Besides, the percentage of stocks releasing dividends is surprisingly stable over time.

Moreover, concerning the variables Grade, the cluster is the one with the smallest percentage of stocks classified as “Buy”, while it has the highest percentage of stocks marked as “Neutral” among all the clusters.

Its main sector is “Industrials”, but we can see from Figure D1 that its composition is diversified, more like Cluster 1 than Cluster 2.

C Simulations

Table C1: Means matrices for simulation

Cluster 1	T1	T2	T3
V1	1.75	1.75	1.75
V2	1.75	1.75	1.75
V3	-0.25	-0.25	-0.25
V4	1	1	1
Cluster 2	T1	T2	T3
V1	2.75	2.75	2.75
V2	2.75	2.75	2.75
V3	0.25	0.25	0.25
V4	2.5	2.5	2.5

D Real data

Table D1: Clusters’ means over time. The estimated parameter $\hat{\pi} = (0.287, 0.156, 0.460, 0.096)$

Cluster 1	2019	2020	2021	2022	2023
Return	19.77	-3.34	28.72	0.03	5.07
Grade	3.93	4.16	4.07	4.07	3.71
Dividend	4.07	4.04	3.44	3.51	3.58
Volume	7.35	7.59	7.38	7.44	7.33
Cluster 2	2019	2020	2021	2022	2023

Return	34.69	31.77	26.73	-27.7	22.21
Grade	3.00	3.24	3.20	3.04	2.69
Dividend	-1.57	-1.92	-2.05	-2.00	-1.68
Volume	5.72	5.82	5.52	5.70	5.66
Cluster 3	2019	2020	2021	2022	2023
Return	27.92	9.54	28.06	-10.87	12.34
Grade	3.09	3.19	3.44	3.23	3.08
Dividend	3.34	3.65	3.58	3.81	4.01
Volume	6.02	6.15	5.91	6.00	5.98
Cluster 4	2019	2020	2021	2022	2023
Return	21.29	22.72	24.87	-43.95	36.45
Grade	4.52	3.71	3.71	3.52	3.46
Dividend	0.50	0.38	-0.88	-0.54	-0.13
Volume	8.04	8.84	8.44	8.52	8.33

Table D2: Clusters' time covariances

Cluster 1	2019	2020	2021	2022	2023
2019	1.36	0.94	0.75	0.61	0.66
2020	0.94	1.64	0.81	0.61	0.67
2021	0.75	0.81	1.51	0.99	0.77
2022	0.61	0.61	0.99	1.68	0.83
2023	0.66	0.67	0.77	0.83	1.3
Cluster 2	2019	2020	2021	2022	2023
2019	2.25	1.97	1.81	1.69	1.79
2020	1.97	2.42	1.94	1.78	1.92
2021	1.81	1.94	2.54	2.02	2.07
2022	1.69	1.78	2.02	2.69	2.2
2023	1.79	1.92	2.07	2.2	2.73
Cluster 3	2019	2020	2021	2022	2023
2019	1.63	1.28	1.2	1.06	1.06
2020	1.28	1.81	1.3	1.09	1.09
2021	1.2	1.3	1.93	1.29	1.27
2022	1.06	1.09	1.29	1.95	1.29
2023	1.06	1.09	1.27	1.29	1.9
Cluster 4	2019	2020	2021	2022	2023
2019	2.61	1.57	1.59	1.5	1.61
2020	1.57	2.06	1.42	1.16	1.27
2021	1.59	1.42	1.95	1.48	1.48

2022	1.5	1.16	1.48	1.92	1.52
2023	1.61	1.27	1.48	1.52	1.97

Table D3: Clusters' variables covariances

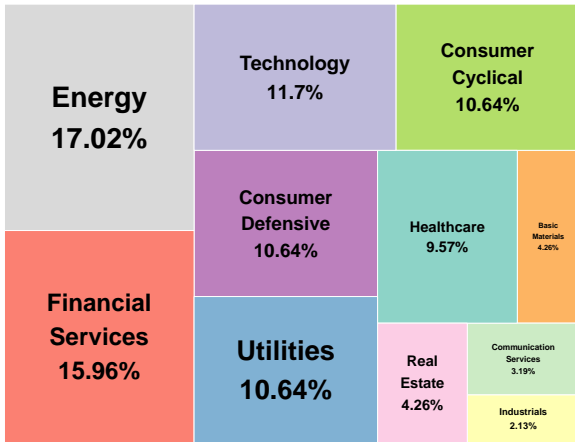
Cluster 1	Return	Grade	Dividend	Volume
Return	589.69	6.4	6.22	-0.87
Grade	6.4	3.36	0.92	-0.06
Dividend	6.22	0.92	3.74	-0.03
Volume	-0.87	-0.06	-0.03	0.1
Cluster 2	Return	Grade	Dividend	Volume
Return	976.02	5.09	3.85	-1.82
Grade	5.09	2.02	0.54	0
Dividend	3.85	0.54	1.98	-0.01
Volume	-1.82	0	-0.01	0.15
Cluster 3	Return	Grade	Dividend	Volume
Return	521.79	1.15	1.89	-0.91
Grade	1.15	3.6	0.97	-0.01
Dividend	1.89	0.97	3.89	-0.01
Volume	-0.91	-0.01	-0.01	0.07
Cluster 4	Return	Grade	Dividend	Volume
Return	2378.8	4.74	-8.1	1.2
Grade	4.74	2.64	0.61	-0.05
Dividend	-8.1	0.61	2.31	0.04
Volume	1.2	-0.05	0.04	0.32

Table D4: Stocks' tickers in each cluster

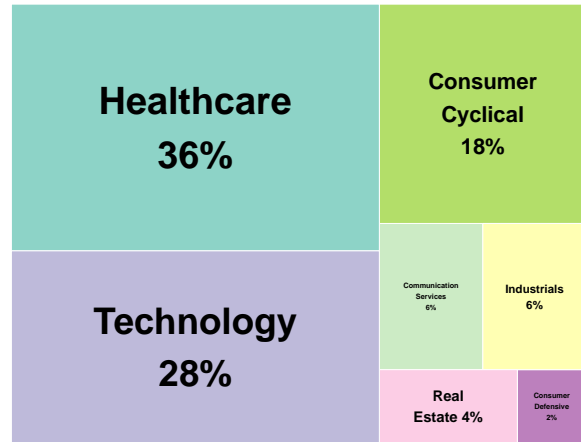
Cluster 1	Cluster 2	Cluster 3	Cluster 4
ABBV, AES, AIG	ADBE, ADSK, ANET	A, ACGL, ACN	AAL, AAPL, AMD
AMAT, BK, BKR	APTV, AZO, BKNG	ADI, ADP, AEE	AMZN, AVGO, BA
BMJ, BX, CFG	BSX, CBRE, CDNS	ALB, ALL, AME	CMG, CRWD, CZR
CL, CMCSA, CNP	CNC, CRL, CSGP	APD, AVB, AVY	DAL, DIS, EXPE
COP, CSCO, CSX	CTLT, DECK, DLTR	AWK, AXP, BALL	F, FCX, GM
CVS, CVX, D	DVA, DXCM, EPAM	BAX, BBY, BEN	GOOGL, INTC, MRNA
DD, DOW, DVN	EW, FFIV, FSLR	BWA, BXP, CAT	MSFT, NCLH, NFLX
EBAY, EOG, EXC	FTNT, GNRC, HOLX	CBOE, CDW, CE	NVDA, PCG, PFE
FANG, FE, FIS	HSIC, IDXX, IQV	CF, CHD, CHRW	PYPL, RCL, SPG
FITB, FOXA, GILD	ISRG, IT, KMX	CLX, CME, CMI	T, TSLA, UAL
GLW, HAL, HBAN	LH, LULU, MHK	CMS, COST, CPB	UBER, WDC

Cluster 1	Cluster 2	Cluster 3	Cluster 4
HD, HPE, HPQ	MOH, MTCH, MTD	CPT, CTAS, CTSH	
IBM, IVZ, JPM	NOW, NVR, ORLY	DE, DFS, DGX	
KDP, KHC, KIM	PANW, PAYC, PTC	DHI, DOV, DPZ	
KMI, KR, LLY	QRVO, TDG, TMUS	DRI, DTE, DUK	
LOW, LUV, LVS	TTWO, URI, VRTX	EA, ED, EFX	
MCHP, MDLZ, MDT	WAT, WST	EIX, EL, ELV	
MET, MGM, MO		EMN, EMR, EQR	
MOS, MPC, MRK		ES, ESS, ETN	
MRO, NEE, NEM		ETR, EVRG, EXR	
NI, NKE, O		FDS, FDX, FMC	
OKE, ORCL, OXY		FTV, GD, GPC	
PEP, PG, PM		GS, HCA, HES	
PPL, QCOM, RF		HII, HON, HRL	
SBUX, SCHW, SLB		HSY, HUM, ICE	
SO, SYF, TGT		INTU, IP, IRM	
TJX, TPR, TXN		ITW, JBHT, JBL	
UNH, USB, V		JNPR, K, KKR	
VICI, VLO, VST		KLAC, KMB, LEN	
VZ, WBA, WFC		LMT, LNT, LRCX	
WMB, WY, WYNN		LW, LYB, MA	
XOM		MAS, MCD, MCK	
		MLM, MMC, MMM	
		NDAQ, NRG, NSC	
		NTAP, NTRS, NUE	
		NXPI, ODFL, PAYX	
		PCAR, PEG, PH	
		PHM, PKG, PLD	
		PNC, PNW, PPG	
		PSA, RL, ROK	
		RSG, SBAC, SHW	
		SNA, SRE, STLD	
		STT, STZ, SWK	
		SWKS, SYY, TAP	
		TER, TMO, TRGP	
		TROW, TRV, TSCO	
		TSN, TXT, UDR	
		UHS, UNP, UPS	
		VMC, VRSK, VTR	
		WAB, WEC, WELL	
		WM, WRB, XEL	

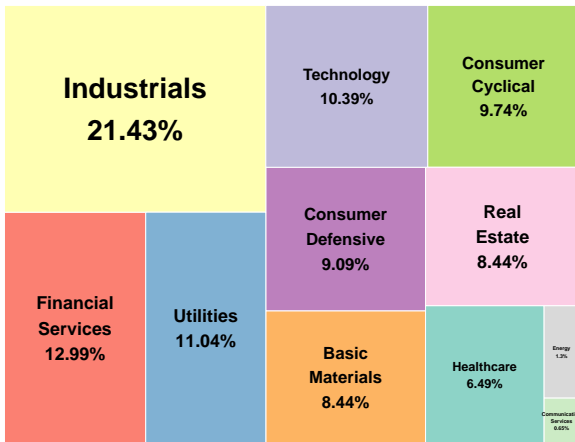
Cluster 1	Cluster 2	Cluster 3	Cluster 4
		ZTS	



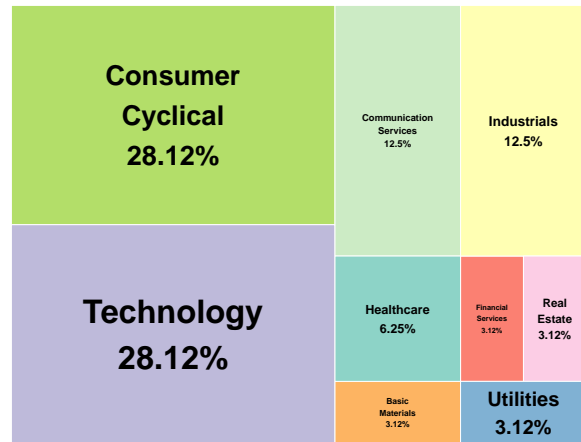
(a) Cluster 1



(b) Cluster 2



(c) Cluster 3



(d) Cluster 2

Figure D1: Clusters' sectors composition

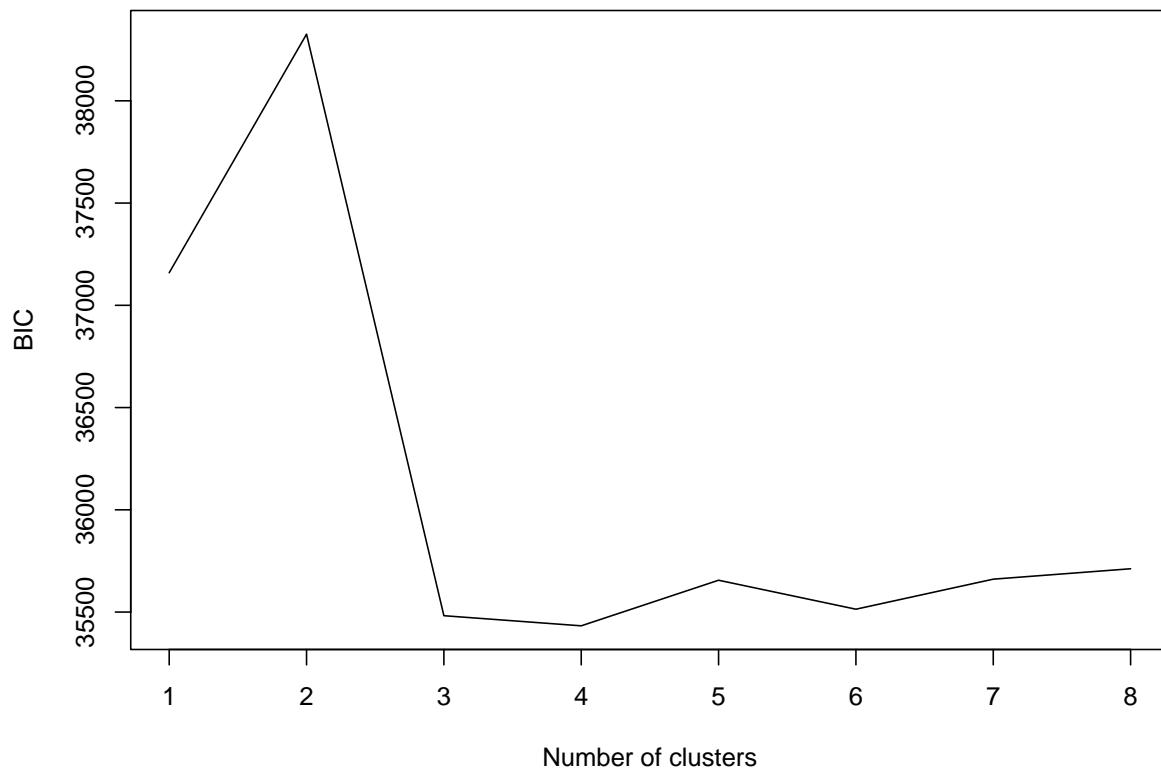


Figure D2: Visualization of BIC for K as results of application on real data. Kmeans++ initialization.