Thesis, Double BSc in Econometrics and Economics
Deep GMM with EUCovS: A layered approach on GMM with hierarchical latent variable concepts

Name Student: Cem Sirin
Student ID: 500707

Thesis Supervisor: dr. C. Cavicchia
Second Assessor: N.W. Koning
Date final version: 3rd of July, 2022

Deep GMM with EUCovS: A layered approach on GMM with hierarchical latent variable concepts

Cem Şirin

July 3, 2022

Abstract

In recent years, there have been numerous new research regarding cluster analysis. This paper combines two existing novel methodologies to form a new class of GMM models: Deep GMMEUCovS. The existing models under scrutiny are Deep GMM (Viroli and McLachlan, 2019) (a layered approach to GMMs) and GMMEUCovS (Cavicchia et al., 2022) (forms GMMs with hierarchical latent variables). Deep GMMEUCovS strives to incorporate these two qualities while exploring other statistical concepts such as multiple cluster structures. The models are applied to popular datasets such as coffee, olive and wine, and later results are compared. A cluster analysis performed on development measures of countries as an exemplary study to highlight the new model's capabilities and shortcomings.

Contents

1 Introduction

Cluster analysis aims to group data with similar properties into clusters. There are many popular clustering techniques available that are based on Partition (K-means and variants), Density, Graph Theory and more (Xu and Tian, 2015). In recent years, there have been many advancements in model-based approaches. Amongst those approaches, Mixture Models attract attention due to their performance and statistical interpretation advantages.

This paper focuses on two particular recent developments concerning Gaussian Mixture Models (GMM). The first one is GMMs with Extended Ultrametric Covariance Structures (EUCovS) (Cavicchia et al., 2022), which offers a novel approach on covariance parametrization that reduces model complexity and tailored towards hierarchical latent structures. The second one is Deep GMMs (Viroli and McLachlan, 2019), that implements the multi-layered structures of deep learning models to GMMs.

GMMs assume that an observation is generated by one of the many Gaussian distributions. In mathematical terms, let $\mathbf{x}_i$ be the i'th observation of a p-dimensional variable for $i = 1, ..., n$. Then, $\mathbf{x}_i$ belongs to a finite Gaussian Mixture Model (GMM) if

$$f(\mathbf{x}_i; \boldsymbol{\theta}) = \sum_{g=1}^{G} \pi_g \phi\left(\mathbf{x}_i; \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right), \tag{1}$$

where index $g = 1, ..., G$ denotes the g'th cluster. $\boldsymbol{\theta}$ is the set of all parameters containing the $[p \times 1]$ mean vector $\boldsymbol{\mu}_g$ and the $[p \times p]$ covariance matrix $\boldsymbol{\Sigma}_g$. $\phi(\cdot)$ denotes the Gaussian distribution, π_g is the probability of an observation appearing in cluster g. A general linear notation for GMM is

$$\mathbf{x}_i = \boldsymbol{\mu}_g + A_g \mathbf{z}_i + \epsilon_i \quad \text{with prob. } \pi_g, \tag{2}$$

where A_g is a $[p \times r]$ transformation matrix, $\mathbf{z}_i$ represents an r-dimensional multivariate normal latent variable, and ϵ_i is the error term which also follows a multivariate normal distribution..

In the following sections, GMMs with EUCovS and Deep GMMs will be introduced and minor adaptations. Onwards, a combination of the two will be introduced and its estimation algorithm. The paper follows up with an application study and insights into the results. Lastly, we will discuss limitations and further areas of research.

2 Notation and theoretical framework

Before introducing the models, it is worth mentioning the challenges of using GMMs, and how previous models have dealt with them. First of all, model complexity is an important issue to address. An typical GMM with G cluster contains $G - 1 + Gp + Gp(p + 1)/2$ parameter estimations. Namely, $G - 1$ mixing proportion, Gp mean vector, $Gp(p + 1)/2$ covariance matrix estimates. As it stands, the number of parameter estimates that belong to the covariance matrices increases polynomially with the number of parameters. GMMs are not known to perform well in high dimensions, therefore many have suggested alternative ways of modelling which leads to reducing number of parameter estimates.

The Parsimonious GMM family later reclassified as the expanded Parsiomious GMM (EPGMM) family (McNicholas and Murphy, 2010) is an important standard in the field and is also used in Deep GMM. Briefly, it assumes $\mathbf{z}_i \sim N(0, I_r)$ which effectively turns equation 2 into a Factor Analysis (FA)

model. Therefore the primary goal here is to fit independent distributed standard normal distributions to describe the observations as best as possible. The GMM with EUCovS, on the other hand, groups variables into latent variables such that they fit an hierarchical structure, functionally laying out the hierarchical structure of the variables. Not all sets of variables have latent hierarchical structures, and the GMMEUCovS is meant to be used in such situations. Thus, one of the focuses of this study is to compare the applications of Deep GMMs by using FA vs EUCovS. We will be inspecting the situations where EUCovS is better and what are its implications.

Another focus of our study is to compare Deep GMMEUCovS with GMMEUCovS. We want to investigate the effects of the extra flexibility given to the estimation process (by being able to extra layers). Finally, there is another concept which is not very popular that this paper would like to address. Multiple cluster structure is the existence of multiple sets of clusters to which an observation can belong. To illustrate, say we have data on the chemical composition. Clusters can form depending on their species (Arabica, Robusta) or depending on where it is produced based on their continent or altitude. We define the *strength* of a cluster set depending on how well it can characterize the data. It is reasonable to expect models that allow multiple cluster structures to fit better when there are cluster sets that are similar in strength. Naively, we may hope that every layer in a Deep GMM can represent a different cluster set, thus allowing for multiple cluster structures. This idea will be followed up in the discussion section explaining what is obtained in our models and could be done to better discover multiple cluster structures.

2.1 GMM with EUCovS

This section introduces the work of Cavicchia et al. (2022), explaining the advantages of GMM with EUCovS, and adapting the model into a linear form. An EUCovS must obey two sets of rules. The first is the properties of a covariance matrix, and the second is the properties of ultrametric matrices laid out by Dellacherie et al. (2014). Given that $\mathbf{\Sigma}$ is a covariance matrix of order p, the following must hold:

(i) Symmetry: $\sigma_{ij} = \sigma_{ji}$ for $i, j = 1, ..., p$,

(ii) Non-negative diagonals: $\sigma_{ii} \geq 0$ for $i = 1, ..., p$,

(iii) Positive semidefiniteness: $\mathbf{x}'\mathbf{\Sigma}\mathbf{x} \geq 0$ for $\forall \mathbf{x} \in \mathbb{R}^p$.

Furthermore for matrix $\mathbf{\Sigma}$ to be considered ultrametric, the following must hold:

(iv) Ultrametric inequality: $\sigma_{ij} \geq \min\{\sigma_{ik}, \sigma_{jk}\}$ for $i, j, k = 1, ..., p$,

(v) Column pointwise diagonal dominance: $\sigma_{ii} \geq \max\{|\sigma_{ij}|, j = 1, ..., p\}$ for $i = 1, ..., p$.

To achieve such structure, every variable is partitioned into groups $q = 1, \ldots, Q$. $\mathbf{V}$ is a $[p \times Q]$ binary membership matrix. Each row is a vector that has a value of 1 on the q'th column representing the relative variable's membership. Cavicchia et al. (2022) presents the following structure to achieve an ultrametric covariance matrix,

$$\mathbf{\Sigma}_{\mathrm{u}} = \mathbf{V}\left(\mathbf{\Sigma}_{\mathrm{W}} + \mathbf{\Sigma}_{\mathrm{B}}\right)\mathbf{V}' - \mathrm{diag}\left(\mathbf{V}\mathbf{\Sigma}_{\mathrm{W}}\mathbf{V}'\right) + \mathrm{diag}\left(\mathbf{V}\mathbf{\Sigma}_{\mathbf{V}}\mathbf{V}'\right), \tag{3}$$

where $\mathbf{\Sigma_V}$ is a $[Q \times Q]$ diagonal matrix where diagonal value q represents the variable group q, $\mathbf{\Sigma_W}$ a $[Q \times Q]$ diagonal matrix where diagonal value q represents the covariance of variable group q, and $\mathbf{\Sigma_B}$

a $[Q \times Q]$ off-diagonal matrix where off-diagonal value q_1, q_2 represents the covariance between variable groups q_1 and q_2. The following constraints makes sure that $\mathbf{\Sigma}_{\text{u}}$ is an ultrametric covariance matrix:

$$ {}_B\sigma_{qh} \geq \min\{{}_B\sigma_{qs}, {}_B\sigma_{hs}\} \quad q, h, s = 1, \ldots, Q, \quad q \neq h \neq s \tag{4} $$

$$ \min\{{}_W\sigma_{qq} : q = 1, \ldots, Q\} \geq \max\{{}_B\sigma_{qh} : q, h = 1, \ldots, Q, h \neq q\} \tag{5} $$

$$ {}_V\sigma_{qq} \geq \max\{|{}_W\sigma_{qq}|, |{}_B\sigma_{qh}|, h = 1, \ldots, Q, h \neq q\} \quad q = 1, \ldots, Q \tag{6} $$

$$ \mathbf{\Sigma}_{\text{u}} = \mathbf{\Sigma}_{\text{u}} + a\mathbf{I}_p, \text{ with } a > 0, \text{ and such that } \mathbf{\Sigma}_{\text{u}} \text{ is positive definite,} \tag{7} $$

A linear formulation of a GMMEUCovS can be achieved when,

$$ \mathbf{x}_i = \boldsymbol{\mu}_g + \mathbf{V}_g\mathbf{z}_i + \epsilon_i \quad \text{with prob. } \pi_g, \tag{8} $$

where $\text{Var}(\mathbf{z}_i) = \mathbf{\Sigma}_{\text{W}} + \mathbf{\Sigma}_{\text{B}}$ and $\text{Var}(\epsilon_i) = \text{diag}\,(\mathbf{V}(\mathbf{\Sigma}_{\text{V}} - \mathbf{\Sigma}_{\text{W}})\mathbf{V}')$.

2.2 Deep GMM

Deep GMMs (Viroli and McLachlan, 2019) and extending the GMM with additional layers. Following Equation 2, an H-layer Deep GMM is as follows:

$$ \begin{aligned} \mathbf{x}_i &= \boldsymbol{\mu}_g^{(1)} + A_g^{(1)}\mathbf{z}_i^{(1)} + \epsilon_i^{(1)} \quad \text{with prob. } \pi_{g_1}^{(1)} \\ \mathbf{z}_i^{(1)} &= \boldsymbol{\mu}_g^{(2)} + A_g^{(2)}\mathbf{z}_i^{(2)} + \epsilon_i^{(2)} \quad \text{with prob. } \pi_{g_2}^{(2)} \\ &\ldots \\ \mathbf{z}_i^{(H-1)} &= \boldsymbol{\mu}_g^{(H)} + A_g^{(H)}\mathbf{z}_i^{(H)} + \epsilon_i^{(H)} \quad \text{with prob. } \pi_{g_H}^{(H)}. \end{aligned} \tag{9} $$

with clusters $g_h = 1, \ldots, G_h$ for layer $h = 1, \ldots, H$. To further clarify the properties of the model, in total there are $\sum_{h=1}^{H}\sum_{g_h=1}^{G_h} g_h$ clusters in the entire system. Observation $\mathbf{x}_i$ is characterised by H clusters its latent variables belong to. In total, there are $\prod_{h=1}^{H} G_h$ unique cluster combinations or routes which is visualized in Figure X. For ease of notation, we will assume that $\mathbf{x}_i = \mathbf{z}_i^{(0)}$.
The distribution of variable $\mathbf{z}_i^{(h)}$ depends on the latent variables that are deeper in the model, i.e., $\mathbf{z}_i^{(h+1)}, \ldots, \mathbf{z}_i^{(H)}$. Like aforementioned, the variable $\mathbf{z}_i^{(h)}$ has $\prod_{l=1}^{h} G_l$ combinations or routes that characterises itself. We will use the symbol $s_h^{(h)} = 1, \ldots, S_h$ to to index these combinations for layer h, and $S_h = \prod_{l=1}^{h} G_l$. We can alternatively write the pdf as

$$ f(\mathbf{z}_i^{(h)}; \mathbf{\Theta}) = \sum_{s_h=1}^{S_h} \pi_{s_h}\phi\left(\mathbf{z}_i^{(h+1)}; \boldsymbol{\mu}_{s_h}, \mathbf{\Sigma}_{s_h}\right) \tag{10} $$

3 Deep GMM with EUCovS

After all the prep work, combining the two model is pretty straight forward. A Deep GMMEUCovS with H layers can be formulized simply by joining equations 8 and 9:

$$\begin{aligned}
\mathbf{x}_i &= \boldsymbol{\mu}_{g_1}^{(1)} + \mathbf{V}_{g_1}^{(1)}\mathbf{z}_i^{(1)} + \epsilon_i^{(1)} \quad \text{with prob. } \pi_{g_1}^{(1)} \\
\mathbf{z}_i^{(1)} &= \boldsymbol{\mu}_{g_2}^{(2)} + \mathbf{V}_{g_2}^{(2)}\mathbf{z}_i^{(2)} + \epsilon_i^{(2)} \quad \text{with prob. } \pi_{g_2}^{(2)} \\
&\dots \\
\mathbf{z}_i^{(H-1)} &= \boldsymbol{\mu}_{g_H}^{(H)} + \mathbf{V}_{g_H}^{(H)}\mathbf{z}_i^{(H)} + \epsilon_i^{(H)} \quad \text{with prob. } \pi_{g_H}^{(H)}.
\end{aligned} \tag{11}$$

where the numerical superscripts signify the index of the layer. For ease of notation, we will assume that $\mathbf{x}_i = \mathbf{z}_i^0$. In given layer h, we iteratively replace the latent variables until we get to the final layer H,

$$\begin{aligned}
\mathbf{z}_i^{(h-1)} &= \boldsymbol{\mu}_{g_h}^{(h)} + \mathbf{V}_{g_h}^{(h)}\mathbf{z}_i^{(h)} + \epsilon_i^{(h)} \quad \text{with prob. } \pi_{g_h}^{(h)} \\
\mathbf{z}_i^{(h-1)} &= \boldsymbol{\mu}_{g_h}^{(h)} + \mathbf{V}_{g_h}^{(h)}\left(\boldsymbol{\mu}_{g_{h+1}}^{(h+1)} + \mathbf{V}_{g_{h+1}}^{(h+1)}\mathbf{z}_i^{(h+1)} + \epsilon_i^{(h+1)}\right) + \epsilon_i^{(h)} \quad \text{with prob. } \pi_{g_h}^{(h)} \times \pi_{g_{h+1}}^{(h+1)} \\
&\dots \\
\mathbf{z}_i^{(h-1)} &= \boldsymbol{\mu}_{g_h}^{(h)} + \sum_{k=h+1}^{H}\left(\prod_{l=k}^{H}\mathbf{V}_{g_l}^{(l)}\right)\boldsymbol{\mu}_{g_l}^{(l)} + \left(\prod_{l=h}^{H}\mathbf{V}_{g_l}^{(l)}\right)\mathbf{z}_{g_H}^{(H)} + \epsilon_i^{(h)} + \sum_{k=h+1}^{H}\left(\prod_{l=k}^{H}\mathbf{V}_{g_l}^{(l)}\right)\epsilon_i^{(l)} \quad \text{with prob. } \prod_{l=h}^{H}\pi_{g_l}^{(l)} \\
\mathbf{z}_i^{(h-1)} &= \widetilde{\boldsymbol{\mu}}_{s_h}^{(h)} + \widetilde{\mathbf{V}}_{s_h}^{(h)}\mathbf{z}_{g_H}^{(H)} + \tilde{\epsilon}_i^{(h)} \quad \text{with prob. } \tilde{\pi}_{s_l}^{(h)}.
\end{aligned} \tag{12}$$

where $\widetilde{\mathbf{V}}_{s_h}^{(h)} = \left(\prod_{l=h}^{H}\mathbf{V}_{g_l}^{(l)}\right)$, $\widetilde{\boldsymbol{\mu}}_{s_h}^{(h)} = \boldsymbol{\mu}_{g_h}^{(h)} + \sum_{l=h+1}^{H}\widetilde{\mathbf{V}}_{s_{l-1}}^{(l-1)}\boldsymbol{\mu}_{g_l}^{(l)}$, $\tilde{\epsilon}_i^{(h)} = \epsilon_i^{(h)} + \sum_{l=h+1}^{H}\widetilde{\mathbf{V}}_{s_{l-1}}^{(l-1)}\epsilon_i^{(l)}$, and $\tilde{\pi}_{s_l}^{(h)} = \prod_{l=h}^{H}\pi_{g_l}^{(l)}$. Thus we can write it as:

$$f(\mathbf{z}_i^{(h-1)};\boldsymbol{\Theta}) = \sum_{s_h=1}^{S_h}\pi_{s_h}^{(h)}\phi\left(\mathbf{z}_i^{(H)};\widetilde{\boldsymbol{\mu}}_{s_h}^{(h)},\widetilde{\boldsymbol{\Sigma}}_{s_h}^{(h)}\right) \tag{13}$$

where $\widetilde{\boldsymbol{\Sigma}}_{s_h}^{(h)} = \widetilde{\mathbf{V}}_{s_h}^{(h)}\boldsymbol{\Sigma}_{g_H}^{(H)}\widetilde{\mathbf{V}}_{s_h}^{(h)} + \sum_{l=h+1}^{H}\widetilde{\mathbf{V}}_{s_{l-1}}^{(l-1)}\left(\boldsymbol{\Sigma}_{\mathrm{V},g_l}^{(l)} - \boldsymbol{\Sigma}_{\mathrm{W},g_l}^{(l)}\right)\widetilde{\mathbf{V}}_{s_{l-1}}^{(l-1)} + \left(\boldsymbol{\Sigma}_{\mathrm{V},g_h}^{(h)} - \boldsymbol{\Sigma}_{\mathrm{W},g_h}^{(h)}\right)$.

4 Estimation

Expectation Maximization (EM) is a common method of estimation of GMMs. Our case requires fitting parameters based on the distribution of the latent variables. Therefore to tackle the analytical and computational complexity of dealing with the distribution of the latent variables, a common approach is to use Stochastic EM (SEM) (Nielsen, 2000). SEM can be summarized in 2 steps. The Expactation Step in EM is replaced with the Stochastic Expectation Step. In this step we, simulate the latent variables given the estimated parameters. Next, we perform the Maximization Step, where we update the parameters given the observed data and the simulated latent variables. To estimate Deep GMMEUCovS, we maximize the following likelihood function.

$$\ell(\boldsymbol{\Psi}) = \sum_{h=1}^{H}\sum_{i=1}^{n}\log\left(\sum_{g_h=1}^{G_h}\pi_{g_h}\phi\left(\mathbf{z}_i^{(h-1)} \mid \mathbf{z}_i^{(h)};\boldsymbol{\Theta}\right)\right) \tag{14}$$

4.1 Stochastic Expectation Step

SEM uses the conditional distribution to estimate the expected value of the latent variables. Since we know $\mathbf{z}^{(0)}$, we start by simulating the first set of latent variables $\mathbf{z}^{(1)}$, then continue simulating the next set of latent variables using the previous. From equation (X), by using some simple matrix algebra we get

$$\mathbf{z}_i^{(h)} = \left(\mathbf{V}_{g_h}^{(h)\prime}\mathbf{V}_{g_h}^{(h)}\right)^{-1}\mathbf{V}_{g_h}^{(h)\prime}\left(\mathbf{z}_i^{(h-1)} - \boldsymbol{\mu}_{g_h}^{(h)} - \epsilon_i^{(h)}\right). \tag{15}$$

Let $\widehat{\mathbf{T}}_{g_h}^{(h)} = \left(\widehat{\mathbf{V}}_{g_h}^{(h)\prime}\widehat{\mathbf{V}}_{g_h}^{(h)}\right)^{-1}\widehat{\mathbf{V}}_{g_h}^{(h)\prime}$, then conditional expected value and variance can be written as the following

$$\begin{aligned}\mathrm{E}\left(\mathbf{z}_i^{(h)} \mid \mathbf{z}_i^{(h-1)}\right) &= \mathrm{E}\left(\sum_{g_h=1}^{G_h}\mathrm{E}\left(\mathbf{z}_i^{(h)} \mid \mathbf{z}_i^{(h-1)}, g_h\right)\mathrm{P}\left(g_h \mid \mathbf{z}_i^{(h-1)}\right)\right)\\ &= \sum_{g_h=1}^{G_h}\widehat{w}_{i,g_h}\widehat{\mathbf{T}}_{g_h}^{(h)}\left(\mathbf{z}_i^{(h-1)} - \widehat{\boldsymbol{\mu}}_{g_h}^{(h)}\right)\end{aligned} \tag{16}$$

$$\begin{aligned}\mathrm{Var}\left(\mathbf{z}_i^{(h)} \mid \mathbf{z}_i^{(h-1)}\right) &= \mathrm{E}\left(\sum_{g_h=1}^{G_h}\mathrm{Var}\left(\mathbf{z}_i^{(h)} \mid \mathbf{z}_i^{(h-1)}, g_h\right)\mathrm{P}\left(g_h \mid \mathbf{z}_i^{(h-1)}\right)\right)\\ &= \sum_{g_h=1}^{G_h}\widehat{w}_{i,g_h}\widehat{\mathbf{T}}_{g_h}^{(h)}\left(\widehat{\boldsymbol{\Sigma}}_{\mathrm{V},g_h}^{(h)} - \widehat{\boldsymbol{\Sigma}}_{\mathrm{W},g_h}^{(h)}\right)\widehat{\mathbf{T}}_{g_h}^{(h)\prime}\end{aligned} \tag{17}$$

Since the linear combination of independent normal distributions is also a normal distribution, $f\left(\mathbf{z}_i^{(h)} \mid \mathbf{z}_i^{(h-1)}\right)$ is also a normal distribution with mean and variance given in equations 16 and 17 respectively. Since $\mathbf{z}_i^{(0)}$ (i.e., the observed data $\mathbf{x}_i$) is known, we start by simulating values $\widetilde{\mathbf{z}}_{i,m}^{(1)}$ and continue deeper into the model up until $\widetilde{\mathbf{z}}_{i,m}^{(H)}$ for $m = 1, \ldots, M$ where M is the number of simulations per variable. We use

$$E\left[\mathbf{z}_i^{(h)} \mid \mathbf{z}_i^{(h-1)}, s; \Theta'\right] \cong \frac{\sum_{m=1}^{M}\mathbf{z}_{i,m}^{(h)}}{M}, \tag{18}$$

which the average of M simulations.

4.2 Maximization Step

We start by estimating the posterior probabilities $w_{i,g_h}^{(h)}$.

$$\widehat{w}_{ig_h}^{(h)} = \frac{\widehat{\pi}_{g_h}^{(h)}\phi\left(\mathbf{z}_i^{(h-1)} \mid \mathbf{z}_i^{(h)}; \widehat{\boldsymbol{\theta}}_{g_h}\right)}{\sum_{g_l=1}^{G_h}\widehat{\pi}_{g_l}^{(h)}\phi\left(\mathbf{z}_i^{(h-1)} \mid \mathbf{z}_i^{(h)}; \widehat{\boldsymbol{\theta}}_{g_l}\right)} \tag{19}$$

We continue with updating the prior probabilities

$$\widehat{\pi}_{g_h}^{(h)} = \frac{\sum_{i=1}^{n}\widehat{w}_{ig_h}^{(h)}}{n} \tag{20}$$

We estimate the mean starting from the deepest layer, iteratively plugging the values in as the following:

$$
\begin{aligned}
\widehat{\boldsymbol{\mu}}_{g_H}^{(H)} &= \frac{\sum_{i=1}^{n} \widehat{w}_{ig_H} \mathbf{z}_i^{(H-1)}}{n_{g_H}} \\
&\cdots \\
\widehat{\boldsymbol{\mu}}_{g_h}^{(h)} &= \frac{\sum_{i=1}^{n} \widehat{w}_{ig_h} \mathbf{z}_i^{(h-1)}}{n_{g_h}} - \sum_{g_l=1}^{G_{h+1}} \hat{\rho}_{g_h,g_l}^{(h)} \widehat{\mathbf{V}}_{g_h} \widehat{\boldsymbol{\mu}}_{g_l}^{(h+1)}
\end{aligned}
\tag{21}
$$

where $\hat{\rho}_{g_h,g_l}^{(h)} = \sum_{i=1}^{n} \widehat{w}_{ig_h}^{(h)} \widehat{w}_{ig_l}^{(h+1)} / n_{g_h}$, namely, the probability of an observation belonging to cluster g_l given that it belongs to g_h. For given $\mathbf{V}$, we estimate the covariance matrices. We start with estimating the covariance matrices of the very final layer H. This part is exactly same as Cavicchia et al. (2022).

$$
\widehat{\boldsymbol{\Sigma}}_{\mathrm{V}_{g_H}}^{(H)} = \left(\widehat{\mathbf{V}}_{g_H}^{(H)\prime} \widehat{\mathbf{V}}_{g_h}^{(H)}\right)^{-1} \widehat{\mathbf{V}}_{g_H}^{(H)\prime} \operatorname{diag}\left(\mathbf{S}_{g_H}^{(H)}\right) \widehat{\mathbf{V}}_{g_H}^{(H)} \quad g_H = 1, \ldots, G_H \tag{22}
$$

where $\mathbf{S}_{g_H}^{(H)}$ is the sample covariance obtained from the simulated latent variables. We estimate covariance within variable groups $\widehat{\boldsymbol{\Sigma}}_{\mathrm{W}_{g_H}}$ and covariance across variable groups $\widehat{\boldsymbol{\Sigma}}_{\mathrm{B}_{g_H}}$ as

$$
\widehat{\boldsymbol{\Sigma}}_{\mathrm{W}_{g_H}}^{(H)} = \left[\left(\widehat{\mathbf{V}}_{g_H}^{(H)\prime} \widehat{\mathbf{V}}_{g_H}^{(H)}\right)^2 - \widehat{\mathbf{V}}_{g_H}^{(H)\prime} \widehat{\mathbf{V}}_{g_H}^{(H)}\right]^{-1} \operatorname{diag}\left[\widehat{\mathbf{V}}_{g_H}^{(H)\prime} \left(\mathbf{S}_{g_H} - \operatorname{diag}\left(\mathbf{S}_{g_H}\right)\right) \widehat{\mathbf{V}}_{g_H}^{(H)}\right] \tag{23}
$$

$$
\widetilde{\boldsymbol{\Sigma}}_{\mathrm{B}g_H}^{(H)} = \widehat{\mathbf{V}}_{g_H}^{(H)+} \mathbf{S}_{g_H} \left(\widehat{\mathbf{V}}_{g_H}^{(H)\prime}\right)^{+} \tag{24}
$$

where then we calculate $\widehat{\boldsymbol{\Sigma}}_{\mathrm{B}_{g_H}}$ by finding the closest matrix to $\widetilde{\boldsymbol{\Sigma}}_{\mathrm{B}_{g_H}}$ in terms of Frobenius distance such that it satisfies equation (3). We then check if $\widehat{\boldsymbol{\Sigma}}_{\mathrm{W}_{g_H}}^{(H)}$ and $\widehat{\boldsymbol{\Sigma}}_{\mathrm{V}_{g_H}}^{(H)}$ meet the constraints 2 and 3, and increase the values that does not fit. Now we have all the ingredient to form the ultrametric covariance matrix:

$$
\widehat{\boldsymbol{\Sigma}}_{g_H}^{(H)} = \widehat{\mathbf{V}}_{g_H}^{(H)} \left(\widehat{\boldsymbol{\Sigma}}_{\mathrm{W}g_H} + \widehat{\boldsymbol{\Sigma}}_{\mathrm{B}g_H}\right) \widehat{\mathbf{V}}_{g_H}^{(H)\prime} + \operatorname{diag}\left(\widehat{\mathbf{V}}_{g_H}^{(H)} \left(\boldsymbol{\Sigma}_{\mathbf{V}_{g_H}}^{(H)} - \boldsymbol{\Sigma}_{\mathrm{W}_{g_H}}^{(H)}\right) \widehat{\mathbf{V}}_{g_H}^{(H)\prime}\right) \tag{25}
$$

Now that we have calculated the variance matrixes for the deepest layer, we can use them to calculate the variance matrixes the layer that is one step shallower and iteratively continute till we reach the first layer.

$$
\begin{aligned}
\widehat{\boldsymbol{\Sigma}}_{\mathrm{V}_{g_h}}^{(h)} &= \left(\widehat{\mathbf{V}}_{g_h}^{(h)\prime} \widehat{\mathbf{V}}_{g_h}^{(h)}\right)^{-1} \widehat{\mathbf{V}}_{g_h}^{(h)\prime} \operatorname{diag}\left(\mathbf{S}_{g_h}^{(h)}\right) \widehat{\mathbf{V}}_{g_h}^{(h)} \\
\widehat{\boldsymbol{\Sigma}}_{\mathrm{W}_{g_h}}^{(h)} &= \left[\left(\widehat{\mathbf{V}}_{g_h}^{(h)\prime} \widehat{\mathbf{V}}_{g_h}^{(h)}\right)^2 - \widehat{\mathbf{V}}_{g_h}^{(h)\prime} \widehat{\mathbf{V}}_{g_h}^{(h)}\right]^{-1} \operatorname{diag}\left[\widehat{\mathbf{V}}_{g_h}^{(h)\prime} \left(\widehat{\boldsymbol{\Sigma}}_{g_h}^{(h+1)} - \operatorname{diag}\left(\widehat{\boldsymbol{\Sigma}}_{g_h}^{(h+1)}\right)\right) \widehat{\mathbf{V}}_{g_h}^{(h)}\right] \\
\widetilde{\boldsymbol{\Sigma}}_{\mathrm{B}g_h}^{(h)} &= \widehat{\mathbf{V}}_{g_h}^{(h)+} \widehat{\boldsymbol{\Sigma}}_{g_h}^{(h+1)} \left(\widehat{\mathbf{V}}_{g_h}^{(h)\prime}\right)^{+},
\end{aligned}
\tag{26}
$$

where $\widehat{\boldsymbol{\Sigma}}_{g_h}^{(h+1)} = \sum_{g_l}^{G_{h+1}} \hat{\rho}_{g_h,g_l}^{(h)} \widehat{\boldsymbol{\Sigma}}_{g_l}^{(h+1)}$.

4.3 Algorithm

- Step 0: Initialize W by k-means and V randomly such that every variable group has at least one variable. Estimate all the parameters.

- Step 1: Simulate z.

- Step 2: Calculate posterior probability W for each layer.

- Step 3: Estimate parameters. If likelihood function (equation 14) has not increased stop, else go back to step 1.

It is important to note that the algorithm does not produce a monotonically increasing likelihood function at every iteration. The function increases sharply in the early iterations, but after it plateaus you may encounter small decreases in the likelihood function. Due to introducing randomness in the generation of the latent variables, there is a chance of having slightly worse model fit after later iterations.

5 Applications

To run the algorithms, we have implemented GMMEUCovS and Deep GMMEUCovS on MATLAB (2022). We have decided to only implement a 2-layered Deep GMMEUCovS, due computational difficulties. We used the package Deep GMM (Viroli and McLachlan, 2020) which is available on the software R (R Core Team, 2022). We have used QGIS (QGIS.org, 2022) to map the results on Figures 1 and 3.

5.1 Coffee, Wine, and Olive data sets

Three data sets were chosen to investigate how the models perform with varying amounts of observations and variables.

Table 1: Performance indicators on real data application: Average Rand Index (ARI) and miss-classification rate (m.r.)

				Deep GMM		GMMEUCovS		Deep GMMEUCovS	
	#obs	#vars	classes	ARI	m.r.	ARI	m.r.	ARI	m.r.
Coffee	43	12	Species,	1	0	1	0	1	0
			Continent	-	-	-	-	0.294	0.372
Olive	572	8	Region,	0.997	0.002	0.720	0.091	0.640	0.125
			Area	-	-	-	-	0.131	0.437
Wine	178	27	Type	0.983	0.006	0.930	0.022	0.911	0.034

The first issue which we want to address is multiple cluster structures. In our applications in the coffee and olive data sets we attempted to identify a secondary cluster structure using the Deep GMMEUCovS. The coffee data set had better results than olive, however both of them are far from indicating whether the current state of the model is able to identify multiple cluster structures or whether there even is multiple clusters in the data sets.

In our coffee data set application, according the to Bayesian Information Criterion (BIC), the best model specifications are $G_1 = 3$, $G_2 = 2$, $Q_1 = 3$ and $Q_2 = 2$. The second layer perfectly captures the species of the coffee as so the other models. The first layer has 3 clusters, considering the information at hand, it may be reflecting the continent of origin (Americas, Africa, Asia) of the coffee. If this assumption is true, the model is able to classify 26 observations out of 43, constituting a classification rate of 0.628.

Applying the models to the olive data set, we see that Deep GMM performs best in classifying the region, following GMMEUCovS, and Deep GMMEUCovS. Based on this ranking, we can speculate that, perhaps, the olive data set does not contain a hierarchical latent variable structure. GMMEUCovS obtains the highest BIC score with $G = 3$, $Q = 4$. Unfortunately, it was not feasible to run Deep GMMEUCovS

with varying specifications due to high computation time, thus we opted use $G_1 = 3$, $G_2 = 9$, $Q_1 = 3$ and $Q_2 = 2$. We attempted to capture the Region via the first layer, and the area with the second one.

The current state of the Deep GMMEUCovS is not a good way of classifying coffee samples amongst continents, or olives amongst regions, however, with the correct modifications, it is a sign that multiple cluster structures can be modelled via multi-layered GMMs. We expected worse results for the olive data set, as Areas are only sub divisions of the Region classification. Therefore, it is very likely that the two cluster sets will have similar underlying factors that differentiate the clusters (e.g., altitude, rainfall, etc.).

Models performed similarly in the wine data set. One key takeaway is that the GMMEUCovS performed better in clustering the primary class in both olive and wine datasets compared to Deep GMMEUCovS. We observe that there may be a trade off of clustering the primary class while generating information on a secondary cluster. Again, the specifications that gave the best fit for Deep GMMEUCovS were $G_1 = 3$, $G_2 = 9$, $Q_1 = 3$ and $Q_2 = 2$. Due to increased number of parameters as a result of adding an extra layer, the model compensates it by decreasing the dimensions of the latent variables, i.e., Q_1 and Q_w.

5.2 Measures of Development

The following application of Deep GMMEUCovS is exemplary of its capabilities. The data set under investigation contains 15 variables which indicate development on 194 countries (CIA, 2012) . Namely, those are region, population density, coastline (coast/land ratio), net migration, infant mortality (per 1000 births), GDP per capita, literacy, phone ownership, percentage of arable, permanent crops, remaining land (other), birthrate, deathrate, and sector distribution of agriculture, industry, and service. We ran Deep GMMEUCovS with specifications ranging $G_1 = 1, \ldots, 10$, $G_2 = 1, \ldots, 10$, $Q_1 = j + 1, \ldots, 8$, and $Q_2 = 1, \ldots, j$ where $j = 2, \ldots, 7$. The best model $G_1 = 3$, $G_2 = 4$, $Q_1 = 3$, and $Q_2 = 2$.

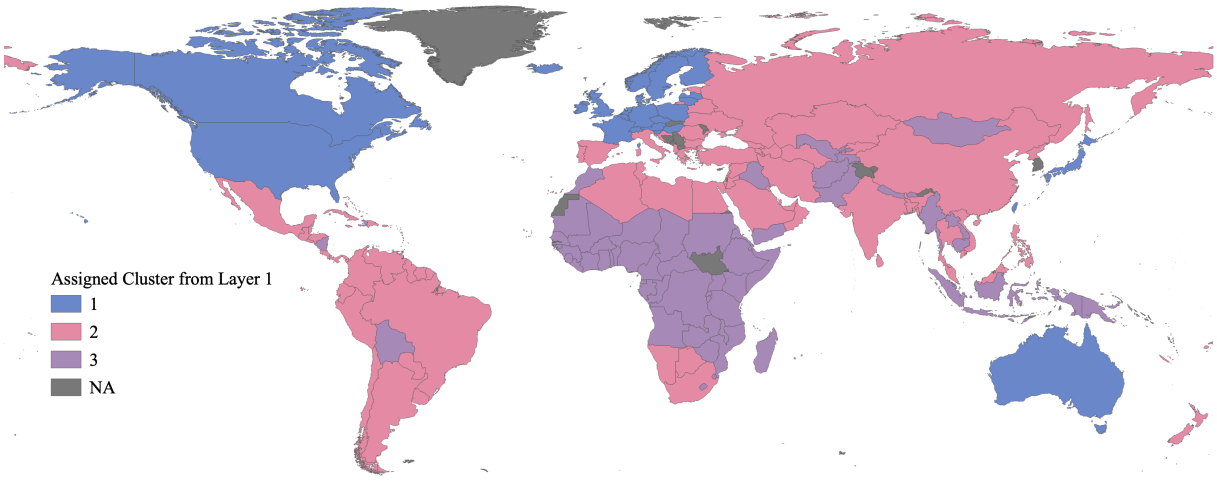

Figure 1: Clustering of countries by development indicators, results produced by layer 1

Figure 1 shows the most likely cluster assignments of the countries in the first layer. Note that

$G_1 = 3$, i.e., there are three different Gaussian functions to which a country can belong. It seems that the countries are partitioned into three categories decreasing levels of development, from 1 to 3. Interesting enough, New Zealand and Southern European countries such as Spain, Italy, and Portugal, which are widely considered developed countries, seem to be assigned to the second tier of countries. In contrast, the Baltic countries Lithuania and Latvia have made the cut to join the first tier of countries.

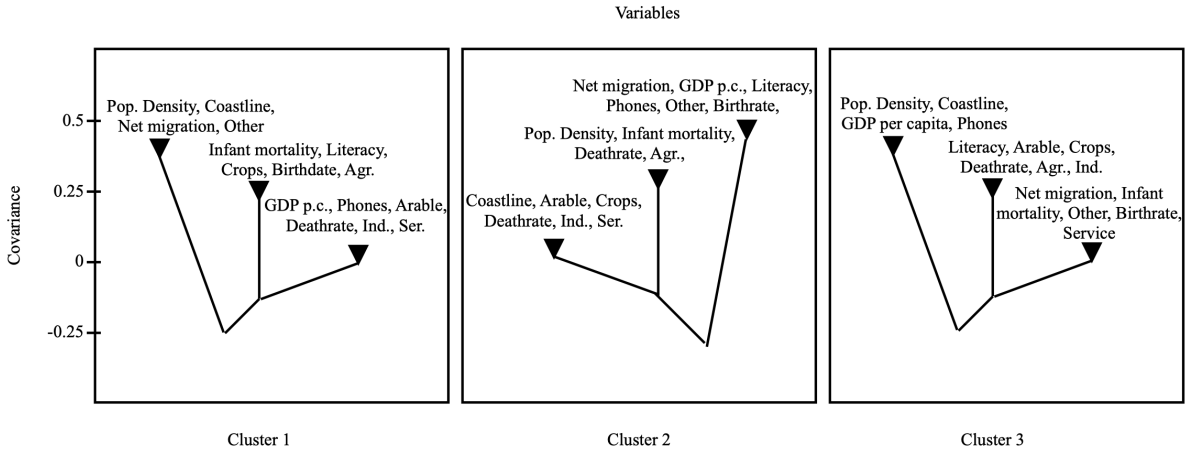

Figure 2: Hierarchical latent variable structure of development indicators, results produced by layer 1.

Figure 2 lays out the hierarchical structure of the variables. The within variable group variances are retrieved from matrix $\widehat{\mathbf{\Sigma}}_{\mathrm{W}g_1}^{(1)}$ and the across variable group variances are retrieved from $\widehat{\mathbf{\Sigma}}_{\mathrm{B}g_1}^{(1)}$. For instance, in the first cluster, which represents the most developed countries, the population density, coastline to land ratio, net migration, and the percentage of remaining land left from agriculture forms the variable group that is at the top of the hierarchy. The variable on the top of the hierarchy has the highest amount of within-group covariance. In the case of the developed countries, population density and coastline ratio are highly correlated since coastal regions are more densely populated[1]. We may speculate that developed countries that receive net migration are also more densely populated, and coasts are more attractive destinations for migrants. Lastly, the land remaining from agriculture may have to increase since land allocated for urban development increases with population. A similar interpretation could be done to the remaining variable groups.

In application, let's assume a policy maker in an underdeveloped country (cluster 3) aims to improve the literacy level of her country. The variable group of which literacy is a part additionally contains the percentage of arable land, percentage of land devoted to permanent crops and share of agriculture. This information implies that agriculture and food-related policies have a strong relationship with literacy levels. Then, the policy maker can further zoom into topics such as food insecurity and distribution, and later the effect on primary education.

Figure 3 shows the most likely cluster assignments of the countries in the second layer. As a reminder, the model specification has $G_2 = 5$, representing five different clusters. It is quite hard to understand the underlying concepts this map represents. Orange countries seem to be fossil fuel-rich counties, and dark green countries are richer in social democracy. However, it is near impossible to say anything coherent

[1]More than 85 per cent of Australians live within 50 kilometres of the sea. (Clark and Johnston, 2017)

with our current tools.

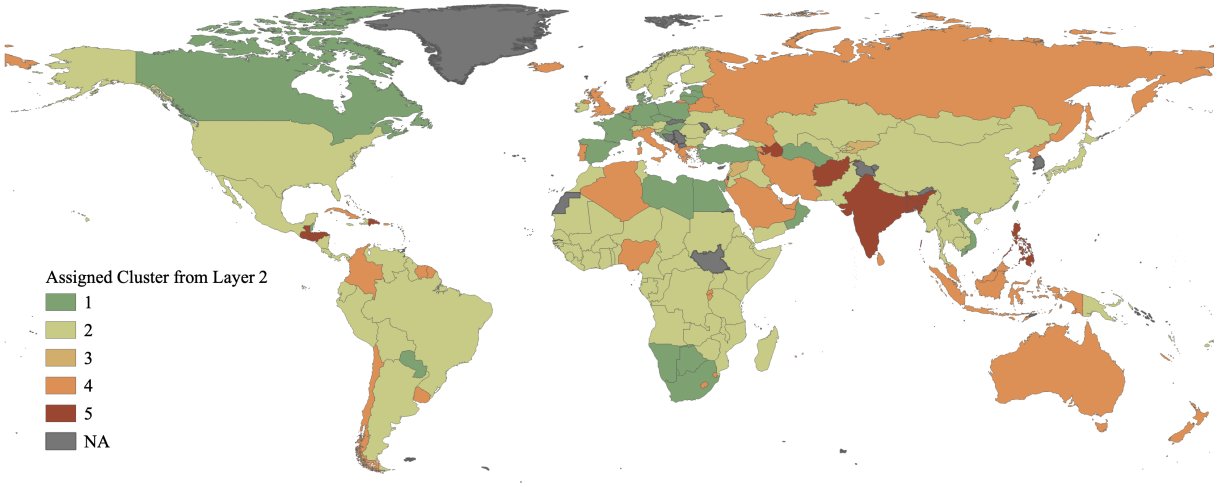

Figure 3: Clustering of countries by development indicators, results produced by layer 2

5.3 Future Development

One downside of the current state of the model is the computation time that it takes to find the best model. There are a lot of combinations of number clusters at every layer $(G_1, \ldots, G_H)$ and number of dimensions at every layer $(Q_1, \ldots, Q_H)$. Moreover, since it is also initialization sensitive, we run it multiple times per given specification. A way to overcome this issue is to set a target number of clusters per layer with some cheap clustering algorithm (such as K-means). Afterwards, search the immediate specifications and continue towards a direction based on some criteria.

An area of improvement, which was not implemented, is to maximize values of within variable group covariance Σ_W. The covariance within the group is direction sensitive and there is no immediate reason to preserve the original direction of the variable. This idea relates to the concept of hierarchical latent variable structures, and how to construct the covariance matrix. Since EUCovS is pretty recent, we still need more applications and trials to have a better idea of what the latent variables conceptualize.

Deep GMMEUCovS prematurely attempted to identify multiple cluster structures. However, the scientific community has not dedicated much attention to the concept of multiple cluster structures. Ideally, we would like to have statistical approaches to measure the existence of multiple cluster structures, and their strength.

References

Cavicchia, C., Vichi, M., and Zaccaria, G. (2022). Gaussian mixture model with an extended ultrametric covariance structure. *Advances in Data Analysis and Classification*, pages 1–29.

CIA, C. I. A. (2012). *The CIA World Factbook 2013*. Central Intelligence Agency. `https://www.cia.gov/the-world-factbook/`.

Clark, G. and Johnston, E. (2017). Australia state of the environment 2016: coasts, independent report to the australian government minister for environment and energy. *Australian Government Department of the Environment and Energy, Canberra*.

Dellacherie, C., Martinez, S., and San Martin, J. (2014). *Inverse M-matrices and ultrametric matrices*, volume 2118. Springer.

MATLAB (2022). *version 9.12.0 (R2022a)*. The MathWorks Inc., Natick, Massachusetts.

McNicholas, P. D. and Murphy, T. B. (2010). Model-based clustering of microarray expression data via latent gaussian mixture models. *Bioinformatics*, 26(21):2705–2712.

Nielsen, S. F. (2000). The stochastic em algorithm: estimation and asymptotic results. *Bernoulli*, pages 457–489.

QGIS.org (2022). *QGIS Geographic Information System*. QGIS Association.

R Core Team (2022). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Viroli, C. and McLachlan, G. J. (2019). Deep gaussian mixture models. *Statistics and Computing*, 29(1):43–51.

Viroli, C. and McLachlan, G. J. (2020). *deepgmm: Deep Gaussian Mixture Models*. R package version 0.1.62.

Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193.