Bayesian Statistics - Final Report

Graphical Models for Categorical Data

Camilla Caroni
Fabio Alberto Comazzi
Andrea Deretti
Francesco Rettore
Michele Russo
Luca Zerman

Tutors: Prof.ssa Lucia Paci, Prof. Federico Castelletti

POLITECNICO
MILANO 1863

Politecnico di Milano
Italy

1 Introduction

1.1 Motivation and outline

The aim of this project is to develop Bayesian methods for the analysis of multivariate categorical data. In particular, we are interested in inferring dependence relations between categorical variables, also accounting for possible heterogeneity related to latent clustering structures in the data.

We adopt graphical models to represent dependence relations between variables: specifically, a *graphical model* is a probabilistic model for a collection of random variables based on a graph structure. A graph $\mathcal{G} = (V, E)$ is made up of a set of *nodes* V (representing variables) and a set of *edges* E (representing dependence relations between nodes). Therefore it can be used to model conditional dependence structures between random variables. In particular, we will focus on *undirected decomposable graphs* (see *Section 1.2* for more details).

We tackle the problem of identifying dependence relations between variables as a model (graph) selection problem. Following a Bayesian perspective, the main ingredients of our model specification are:

- Data Model: $\mathbf{X} = (X_1, \dots X_q) \mid \underline{\theta}, \mathcal{G} \overset{iid}{\sim} p(\underline{x} \mid \underline{\theta}, \mathcal{G})$
- Prior on graph-dependent parameter $\underline{\theta}$ (given the graph): $p(\underline{\theta} \mid \mathcal{G})$
- Prior on graph $\mathcal{G}$: $p(\mathcal{G})$

Target of the analysis is to approximate a posterior distribution over the space of all possible graph structures given the data matrix $\mathbb{X} = \{\underline{x}^{(1)}, \dots, \underline{x}^{(n)}\}$

$$p(\mathcal{G} \mid \mathbb{X}) \propto p(\mathbb{X} \mid \mathcal{G}) p(\mathcal{G}) \tag{1}$$

where in particular

$$p(\mathbb{X} \mid \mathcal{G}) = \int p(\mathbb{X} \mid \underline{\theta}, \mathcal{G}) p(\underline{\theta} \mid \mathcal{G}) d\underline{\theta} \tag{2}$$

is the *marginal likelihood* of graph $\mathcal{G}$ for $\mathcal{G} \in \mathcal{S}_q$, the set of all decomposable graphs on q nodes.

In *Section 2* and *Section 3* we will present two different conjugate models to deal with categorical data, based on:

- **Multivariate categorical (multinomial) distributions** with Hyper Dirichlet prior;
- **Latent multivariate Gaussian distributions** with Hyper-Inverse-Wishart prior (inference via data augmentation).

We will show that in both cases the posterior is known up to a normalizing constant and we will present the implementation (using R) of a Metropolis-Hastings algorithm in order to make inference on $\mathcal{G}$ given the data.

In *Section 4* we consider mixtures of graphical models to account for heterogeneity in the data and perform cluster analysis.

Finally, in *Section 5* we present an assessment of the performances of the two models on a real case dataset, namely the *Congressional Voting Records* dataset; while in *Section 6* we conclude by mentioning some possible future developments regarding our project.

1.2 Notation and graph theory

As mentioned in the previous section, we will focus on undirected decomposable graphs; in particular we will let $\mathcal{G}$ be a decomposable graph and $\mathcal{S}_q$ the space of all decomposable graphs with q nodes.

Definition 1.1. A graph $\mathcal{G}$ is called *decomposable* if each cycle of length $l \geq 4$ admits a chord, that is two non-consecutive adjacent nodes.

An undirected decomposable graph is uniquely determined by its set of *cliques* and *separators*, which are defined as follows:

Definition 1.2. A complete subset which is maximal with respect to inclusion is called a *clique*. We denote by $\mathcal{C}$ the set of all cliques in a graph.

Definition 1.3. Let $\mathcal{C} = \{C_1, \dots C_K\}$. For $k = 2, \dots K$ we define *separators* of the graph the sets

$$S_k = C_k \cap H_{k-1}$$

where $H_{k-1} = C_1 \cup \cdots \cup C_{k-1}$. We denote by $\mathcal{S}$ the set of all separators in a graph.

An important property of decomposable graphs (largely exploited in the following) is that the density function admits the following *factorization*

$$p(\mathbf{X} \mid \underline{\theta}, \mathcal{G}) = \frac{\prod_{C \in \mathcal{C}} p(\mathbf{X}_C \mid \underline{\theta}_C)}{\prod_{S \in \mathcal{S}} p(\mathbf{X}_S \mid \underline{\theta}_S)}. \tag{3}$$

Notice that, when writing $\mathbf{X}_C$ and $\underline{\theta}_C$, we mean the sub-vector of $\mathbf{X}$ and $\underline{\theta}$ indexed by C (and the same holds also for S).

2 Categorical models with Hyper-Dirichlet priors

2.1 Notation and preliminary steps

In this section we consider a collection of categorical variables $X_1, \dots X_q$ where $X_j \in \mathcal{X}_j$, the *set of levels* of X_j, $\forall j \in \{1, \dots q\}$. Therefore, if we consider a vector of realizations $\underline{x} = (x_1, \dots x_q)$ (also named *configuration*), we have that $\underline{x} \in \mathcal{X} := \bigotimes_{j=1}^{q} \mathcal{X}_j$ and we let $\pi(\underline{x})$ be the probability to observe configuration $\underline{x}$. It follows that

$$\sum_{\underline{x} \in \mathcal{X}} \pi(\underline{x}) = 1. \tag{4}$$

Also, we let $\underline{\theta}$ be the vector collecting all the configuration-probabilities, namely

$$\underline{\theta} := \{\pi(\underline{x}), \underline{x} \in \mathcal{X}\}. \tag{5}$$

This vector represents our graph-dependent parameter in this first model, for which it will be important to define a suitable prior.

2.1.1 Contingency tables

For convenience, we can represent the vector of all configuration-probabilities $\underline{\theta}$ through a *contingency table*. We present here a simple example to clarify the notation.

Example 2.1. Consider two binary categorical variables $X_1 \in \{a, b\}$ and $X_2 \in \{0, 1\}$. The whole set of levels of (X_1, X_2) is then

$$\mathcal{X} = \{\{a, 0\}, \{a, 1\}, \{b, 0\}, \{b, 1\}\}.$$

In this case, the contingency table is expressed by

	0	1
a	$\pi(a, 0)$	$\pi(a, 1)$
b	$\pi(b, 0)$	$\pi(b, 1)$

■

Moreover, given a subset of the q variables, we can define a *marginal contingency table* as follows.

Definition 2.1. Consider a subset of indexes $S \subseteq \{1, \dots q\}$ and let $\underline{x}_S$ be the sub-vector of $\underline{x}$ indexed by S, such that $\underline{x}_S \in \mathcal{X}_S = \bigotimes_{j \in S} \mathcal{X}_j$. The (marginal) probability of configuration $\underline{x}_S$ is then defined as

$$\pi(\underline{x}_S) = \sum_{\underline{x} \in \mathcal{X}} \pi(\underline{x}) \mathbb{1}\{\underline{x}(S) = \underline{x}_S\}.$$

If we now let $\underline{\theta}_S$ the vector of probabilities of each possible configuration in $\mathcal{X}_S$, *i.e.*

$$\underline{\theta}_S = \{\pi(\underline{x}_S), \underline{x}_S \in \mathcal{X}_S\},$$

a *marginal contingency table* can be obtained as a tabular representation of $\underline{\theta}_S$ (similarly as before).

In what follows the set of indexes S will represent a suitable clique or separator of a decomposable graph $\mathcal{G}$.

Example 2.2. Moving back to *Example 2.1*, if we consider $S = \{2\}$ then we can write

$$\pi(0) = \pi(a, 0) + \pi(b, 0),$$
$$\pi(1) = \pi(a, 1) + \pi(b, 1).$$

■

2.1.2 Counts

Consider now a dataset $\mathbb{X}$ consisting of n different realizations of the random vector $\mathbf{X}$, namely $\underline{x}^{(i)}, i = 1, \dots n$. For each configuration $\underline{x} \in \mathcal{X}$ we can count the number of observations that are equal to $\underline{x}$; accordingly, we define the *configuration count* as follows.

Definition 2.2. Let $\underline{x} \in \mathcal{X}$ a configuration of $\mathbf{X}$. The *configuration count* is defined by

$$n(\underline{x}) := \sum_{i=1}^{n} \mathbb{1}\{\underline{x}^{(i)} = \underline{x}\}.$$

Analogously, given $\underline{x}_S \in \mathcal{X}_S$ with $S \subseteq \{1, \dots q\}$, the *marginal configuration count* is defined by

$$n(\underline{x}_S) = \sum_{i=1}^{n} \mathbb{1}\{\underline{x}^{(i)}(S) = \underline{x}_S\} = \sum_{\underline{x} \in \mathcal{X}} n(\underline{x}) \mathbb{1}\{\underline{x}(S) = \underline{x}_S\}.$$

2.1.3 Likelihood

Given the notation introduced in *Section 2.1.1* and *Section 2.1.2* we now write the likelihood as a function of the configuration counts. We can first write the *joint probability function* for one single realization $\underline{x}^{(i)} \in \mathcal{X}$ as

$$\mathbb{P}\{X_1 = x_1^{(i)}, \dots, X_q = x_q^{(i)} \mid \underline{\theta}\} = p(\underline{x}^{(i)} \mid \underline{\theta}) = \prod_{\underline{x} \in \mathcal{X}} \pi(\underline{x})^{\mathbb{1}\{\underline{x}^{(i)} = \underline{x}\}}. \tag{6}$$

Then, assuming independence among the n observations $\underline{x}^{(i)}, i = 1, \dots n$, we can write the *likelihood* as

$$\begin{aligned} p(\underline{x}^{(1)}, \dots, \underline{x}^{(n)} \mid \underline{\theta}) &= \prod_{i=1}^{n} \prod_{\underline{x} \in \mathcal{X}} \pi(\underline{x})^{\mathbb{1}\{\underline{x}^{(i)} = \underline{x}\}} \\ &= \prod_{\underline{x} \in \mathcal{X}} \pi(\underline{x})^{\sum_{i=1}^{n} \mathbb{1}\{\underline{x}^{(i)} = \underline{x}\}} \\ &= \prod_{\underline{x} \in \mathcal{X}} \pi(\underline{x})^{n(\underline{x})} \end{aligned} \tag{7}$$

where the last expression is obtained by exchanging the products and exploiting *Definition 2.2*. Similarly, considering a subset of random variables $S \subseteq \{1, \dots q\}$, it is possible to derive the expression of the *likelihood function* restricted to variables indexed by S

$$\begin{aligned} p(\underline{x}_S^{(1)}, \dots, \underline{x}_S^{(n)} \mid \underline{\theta}) = \prod_{i=1}^{n} p(\underline{x}_S^{(i)} \mid \underline{\theta}) &= \prod_{i=1}^{n} \prod_{\underline{x}_S \in \mathcal{X}_S} \pi(\underline{x}_S)^{\mathbb{1}\{\underline{x}_S^{(i)} = \underline{x}_S\}} \\ &= \prod_{\underline{x}_S \in \mathcal{X}_S} \pi(\underline{x}_S)^{\sum_{i=1}^{n} \mathbb{1}\{\underline{x}_S^{(i)} = \underline{x}_S\}} \\ &= \prod_{\underline{x}_S \in \mathcal{X}_S} \pi(\underline{x}_S)^{n(\underline{x}_S)} \end{aligned} \tag{8}$$

where again we have exchanged the products and exploited *Definition 2.2*.

2.2 Definition of the graphical model

Consider now a decomposable graph $\mathcal{G}$. Since we have multivariate categorical random variables $X_1, \dots X_q$, our dataset consists of $\underline{x}^{(1)}, \dots \underline{x}^{(n)}$, with $\underline{x}^{(i)}$ representing the i-th realization of the random vector $\mathbf{X} = (X_1, \dots X_q)$.

In this framework, we can exploit the *factorization property* defined in *Section 1.2* to express the joint density under the decomposable graph $\mathcal{G}$ of a single realization $\underline{x}^{(i)}$ for each $i = 1, \dots n$, namely

$$p(\underline{x}^{(i)} \mid \underline{\theta}, \mathcal{G}) = \frac{\prod_{C \in \mathcal{C}} p(\underline{x}_C^{(i)} \mid \underline{\theta}_C)}{\prod_{S \in \mathcal{S}} p(\underline{x}_S^{(i)} \mid \underline{\theta}_S)} \tag{9}$$

where $\mathcal{C}$ and $\mathcal{S}$ represent the sets of cliques and separators in graph $\mathcal{G}$.

In the following, we will present all the ingredients we need in order to specify the graphical model, *i.e.* the likelihood and the prior both on graph $\mathcal{G}$ and on the graph-dependent parameter $\underline{\theta}$.

2.2.1 Likelihood

Notice that we can express the likelihood either as $p(\underline{x}^{(1)},\dots\underline{x}^{(n)} \mid \underline{\theta},\mathcal{G})$ or as $p(N \mid \underline{\theta},\mathcal{G})$, where N denotes the *contingency table of counts*, namely

$$N = \{n(\underline{x}), \underline{x} \in \mathcal{X}\}. \tag{10}$$

Therefore, because of the *factorization property* and following Equation (8) we can write the likelihood as

$$\begin{aligned} p(\underline{x}^{(1)},\dots\underline{x}^{(n)} \mid \underline{\theta},\mathcal{G}) = \prod_{i=1}^{n} p(\underline{x}^{(i)} \mid \underline{\theta},\mathcal{G}) &= \frac{\prod_{C\in\mathcal{C}} p(N_C \mid \underline{\theta}_C,\mathcal{G})}{\prod_{S\in\mathcal{S}} p(N_S \mid \underline{\theta}_S,\mathcal{G})} \\ &= \frac{\prod_{C\in\mathcal{C}}\prod_{\underline{x}_C\in\mathcal{X}_C} \pi(\underline{x}_C)^{n(\underline{x}_C)}}{\prod_{S\in\mathcal{S}}\prod_{\underline{x}_S\in\mathcal{X}_S} \pi(\underline{x}_S)^{n(\underline{x}_S)}}. \end{aligned} \tag{11}$$

Coherently with previous notation, here $N_C = \{n(\underline{x}_C), \underline{x}_C \in \mathcal{X}_C\}$ and $N_S = \{n(\underline{x}_S), \underline{x}_S \in \mathcal{X}_S\}$ denote *marginal contingency tables of counts*.

2.2.2 Prior on graph-dependent parameter

First of all, we assume that the prior on $\underline{\theta}$ admits the same factorization over cliques and separators of the sampling distribution

$$p(\underline{\theta} \mid \mathcal{G}) = \frac{\prod_{C\in\mathcal{C}} p(\underline{\theta}_C \mid \mathcal{G})}{\prod_{S\in\mathcal{S}} p(\underline{\theta}_S \mid \mathcal{G})}. \tag{12}$$

A conjugate prior for $\underline{\theta}$ under the decomposable graph $\mathcal{G}$ was proposed by Dawid and Lauritzen (1993) who introduced the *Hyper-Dirichlet* distribution; see also [3]. Specifically, we say that $\underline{\theta}$ follows a *Hyper-Dirichlet* distribution if $\forall C \in \mathcal{C}$ (the same holds also $\forall S \in \mathcal{S}$ with obvious change of notation)

$$p(\underline{\theta}_C \mid \mathcal{G}) = \frac{\Gamma(\sum_{\underline{x}_C\in\mathcal{X}_C} a(\underline{x}_C))}{\prod_{\underline{x}_C\in\mathcal{X}_C} \Gamma(a(\underline{x}_C))} \prod_{\underline{x}_C\in\mathcal{X}_C} \pi(\underline{x}_C)^{a(\underline{x}_C)-1}; \tag{13}$$

equivalently $\underline{\theta}_C \mid \mathcal{G} \sim Dir(A_C)$ where $A_C = \{a(\underline{x}_C), \underline{x}_C \in \mathcal{X}_C\}$ and $Dir(\cdot)$ denotes the Dirichlet distribution.

In addition, $p(\underline{\theta})$ determines a unique Hyper-Dirichlet prior for $\underline{\theta}$ provided that $p(\underline{\theta}_C)$ with $C \in \mathcal{C}$ are *hyperconsistent*, meaning

$$\sum_{\underline{x}_C \supseteq \underline{x}_{C\cap C'}} a(\underline{x}_C) = \sum_{\underline{x}_{C'} \supseteq \underline{x}_{C\cap C'}} a(\underline{x}'_C) \tag{14}$$

for each C, C' such that $C \cap C' \neq \emptyset$.

Notice that if we choose $a(\underline{x}_C) = \frac{a}{|\mathcal{X}_C|}$ for some value $a > 0$, the condition expressed in (14) is satisfied.

2.2.3 Prior on the graph

Remember now that graph $\mathcal{G}$ is unknown and represents the target for our model selection problem. Accordingly, we need to specify a prior over $\mathcal{S}_q$, the space of all decomposable graphs of q nodes. Several choices are possible. In particular, we considered three different possibilities for $p(\mathcal{G})$:

- **Uniform prior:** assigns equal probabilities to all the graphs, *i.e.* $p(\mathcal{G}) = \frac{1}{|\mathcal{S}_q|}$.
- **Binomial prior:** assumes $A_{u,v} \mid \pi \overset{iid}{\sim} Be(\pi)$, $\pi \in (0,1)$ where $A_{u,v}$ is the (u,v)-element of the upper triangular adjacency matrix of $\mathcal{G}$.
- **Beta-Binomial prior:** assumes $A_{u,v} \mid \pi \overset{iid}{\sim} Be(\pi)$, $\pi \sim Beta(a,b)$.

Notice that the last two choices are informative priors that can easily incorporate prior information on the degree of *sparsity* in the graph, by suitably tuning a prior probability of edge inclusion hyperparameter π.

2.2.4 Model specification

Finally, the *Multinomial-Hyper-Dirichlet model* can be written as:

$$\begin{aligned} \mathbb{X} \mid \underline{\theta}, \mathcal{G} \sim p(\underline{x}^{(1)}, \dots \underline{x}^{(n)} \mid \underline{\theta}, \mathcal{G}) = \frac{\prod_{C \in \mathcal{C}} p(N_C \mid \underline{\theta}_C, \mathcal{G})}{\prod_{S \in \mathcal{S}} p(N_S \mid \underline{\theta}_S, \mathcal{G})}; \\ \underline{\theta} \mid \mathcal{G} \sim Hyper-Dirichlet(A); \\ \mathcal{G} \sim p(\mathcal{G}) \end{aligned} \tag{15}$$

where $p(\mathcal{G})$ denotes one of the possible priors for graph $\mathcal{G}$ presented in *Section 2.2.3*.

2.2.5 Posterior on graph-dependent parameter

Given the previous model specification, we can easily show that the prior $p(\underline{\theta} \mid \mathcal{G})$ is conjugate. In particular, for a given clique $C \in \mathcal{C}$, we can express the *posterior density* of the parameter θ_C as follows

$$\begin{aligned} p(\underline{\theta}_C \mid N_C, \mathcal{G}) &\propto p(\underline{\theta}_C \mid \mathcal{G}) p(N_C \mid \underline{\theta}_C, \mathcal{G}) \\ &\propto \prod_{\underline{x}_C \in \mathcal{X}_C} \pi(\underline{x}_C)^{a(\underline{x}_C)-1} \prod_{\underline{x}_C \in \mathcal{X}_C} \pi(\underline{x}_C)^{n(\underline{x}_C)} \\ &= \prod_{\underline{x}_C \in \mathcal{X}_C} \pi(\underline{x}_C)^{a(\underline{x}_C)+n(\underline{x}_C)-1}. \end{aligned} \tag{16}$$

From (17) it is straightforward to derive that

$$\underline{\theta}_C \mid N_C, \mathcal{G} \sim Dir(N_C + A_C) \tag{17}$$

with the usual notation for N_C and A_C. A similar procedure applied on a given separator $S \in \mathcal{S}$ leads to

$$\underline{\theta}_S \mid N_S, \mathcal{G} \sim Dir(N_S + A_S) \tag{18}$$

so that the posterior distribution of $\underline{\theta}$ is such that

$$\underline{\theta} \mid N, \mathcal{G} \sim Hyper-Dirichlet(N + A) \tag{19}$$

where the sum between N and A should be intended as element-wise.

2.2.6 Marginal likelihood

To compute the posterior probability of graph $\mathcal{G}$ given the data $\mathbb{X}$ we need first to derive the *marginal likelihood*; see also *Section 1.1*.

The latter is available in closed form by exploiting again the factorization of the likelihood and the prior under a decomposable graph and because of conjugacy of the Hyper-Dirichlet prior with the sampling distribution. In particular, we can write

$$m(N \mid \mathcal{G}) = \frac{\prod_{C \in \mathcal{C}} m(N_C \mid \mathcal{G})}{\prod_{S \in \mathcal{S}} m(N_S \mid \mathcal{G})} \tag{20}$$

where, for each $\forall C \in \mathcal{C}$ and $\forall S \in \mathcal{S}$

$$\begin{aligned} m(N_C \mid \mathcal{G}) &= \frac{\Gamma(\sum_{\underline{x}_C \in \mathcal{X}_C} a(\underline{x}_C))}{\Gamma(\sum_{\underline{x}_C \in \mathcal{X}_C} a(\underline{x}_C) + n(\underline{x}_C))} \prod_{\underline{x}_C \in \mathcal{X}_C} \frac{\Gamma(a(\underline{x}_C) + n(\underline{x}_C))}{\Gamma(a(\underline{x}_C))}; \\ m(N_S \mid \mathcal{G}) &= \frac{\Gamma(\sum_{\underline{x}_S \in \mathcal{X}_S} a(\underline{x}_S))}{\Gamma(\sum_{\underline{x}_S \in \mathcal{X}_S} a(\underline{x}_S) + n(\underline{x}_S))} \prod_{\underline{x}_S \in \mathcal{X}_S} \frac{\Gamma(a(\underline{x}_S) + n(\underline{x}_S))}{\Gamma(a(\underline{x}_S))} \end{aligned} \tag{21}$$

where, using the relations $\sum_{\underline{x}_C \in \mathcal{X}_C} a(\underline{x}_C) = a$ and $\sum_{\underline{x}_C \in \mathcal{X}_C} a(\underline{x}_C) + n(\underline{x}_C) = a + n$ to respect the *Hyper-consistency*, as discussed in *Section* 2.2.2, it is possible to express the *marginal likelihoods* as

$$\begin{aligned} m(N_C \mid \mathcal{G}) &= \frac{\Gamma(a)}{\Gamma(a+n)} \prod_{\underline{x}_C \in \mathcal{X}_C} \frac{\Gamma(a(\underline{x}_C) + n(\underline{x}_C))}{\Gamma(a(\underline{x}_C))}; \\ m(N_S \mid \mathcal{G}) &= \frac{\Gamma(a)}{\Gamma(a+n)} \prod_{\underline{x}_S \in \mathcal{X}_S} \frac{\Gamma(a(\underline{x}_S) + n(\underline{x}_S))}{\Gamma(a(\underline{x}_S))}. \end{aligned} \tag{22}$$

Such closed-form expression of the *marginal likelihood* is crucial to compute the acceptance probability in the Metropolis-Hastings algorithm described in the next section.

2.3 Metropolis-Hastings algorithm

Recall now the posterior distribution of graph $\mathcal{G}$, given the data $\mathbb{X}$, introduced in *Section* 1.1, which can be written as

$$p(\mathcal{G} \mid N) \propto m(N \mid \mathcal{G}) p(\mathcal{G}).$$

Since, however, the dimension of the space $\mathcal{S}_q$ grows super-exponentially with respect to the number of nodes q [1], an exhaustive search of such space is not feasible. Therefore, we need to sample from the posterior distribution $p(\mathcal{G} \mid N)$ by resorting to a Metropolis-Hastings scheme.

2.3.1 Proposal distribution

The first step to implement the Metropolis-Hastings algorithm is to define a suitable *proposal distribution* $q(\mathcal{G}' \mid \mathcal{G})$ from which we sample a new proposal graph ($\mathcal{G}'$) starting from the current state of the chain ($\mathcal{G}$).

[1] Denoting by $n_q = \sum_{i=1}^{q-1} i = \frac{q(q-1)}{2}$ the number of entries in the (upper) triangular part of the adjacency matrix of $\mathcal{G}$, then the number of all possible undirected graphs with q nodes is given by 2^{n_q}. Notice that this number takes into account also the *non decomposable* graphs. For this reason the dimesion of the considered space (*decomposable* and *undirected*) is less than 2^{n_q} but it grows super-exponentially nonetheless.

In particular, we build $\mathcal{G}'$ by either adding (a) or removing (b) an edge from $\mathcal{G}$ according to the following scheme:

1. Construct the space $\mathcal{O}_{\mathcal{G}}$ of all possible *undirected* graphs obtained by (a) or (b).
2. Uniformly draw a graph $\mathcal{G}'$ from $\mathcal{O}_{\mathcal{G}}$.
3. If $\mathcal{G}'$ is decomposable propose it, otherwise go back to step (2).

Notice that such approach is computationally efficient as it guarantees that $\frac{q(\mathcal{G}|\mathcal{G}')}{q(\mathcal{G}'|\mathcal{G})} = 1$, so that we do not need to compute such ratio explicitly when computing the acceptance rate α (see the pseudo-code in *Section 2.3.2*).

2.3.2 Implementation of the algorithm

The Metropolis-Hastings algorithm to sample from the posterior distribution $p(\mathcal{G} \mid N)$ is implemented as follows:

Algorithm MH algorithm for the Multinomial-Hyper-Dirichlet Model

Input: $\mathcal{G}^{(0)}$ (the initial candidate graph), M (the number of MCMC iterations)
Output: An MCMC sample $\{\mathcal{G}^{(t)}\}_{t=1}^{M}$ from $p(\mathcal{G} \mid N)$
for $t \leftarrow 1$ **to** M **do**

$\mathcal{G} \leftarrow \mathcal{G}^{(t-1)}$;

$\mathcal{G}' \sim q(\mathcal{G}' \mid \mathcal{G})$;

compute $\alpha(\mathcal{G}' \mid \mathcal{G}) = \min\left\{1, \frac{m(N|\mathcal{G}')}{m(N|\mathcal{G})} \cdot \frac{p(\mathcal{G}')}{p(\mathcal{G})}\right\}$;

update $\mathcal{G}^{(t)} = \begin{cases} \mathcal{G}', & \text{with probability } \alpha \\ \mathcal{G}^{(t-1)}, & \text{with probability } 1-\alpha \end{cases}$

end

3 Latent Gaussian variables with Hyper-Inverse-Wishart prior

Our second model is still based on a collection of categorical variables $X_1, \dots X_q$, that for simplicity we assume to be all binary, namely $X_j \in \{0,1\}$ for all $j = 1, \dots q$, and a decomposable graph structure $\mathcal{G} \in \mathcal{S}_q$. However, differently from the previous model formulation that we based on Multinomial-Dirichet distributions, we here assume that observed categorical variables are generated by *discretizaion* of latent Gaussian random variables $Z_1, \dots Z_q$.

Accordingly, we organize this section into three main parts: in *Section 3.1* we specify a graphical model formulation for a Gaussian random vector $\mathbf{Z} = (Z_1, \dots Z_q)$ and present a few useful results both in terms of posterior distribution of model parameters and marginal likelihood. In *Section 3.2* we then introduce the likelihood writing explicitly the relationship between latent and observed variables based on *discretization* of the underlying Gaussian data. Finally, we present the Metropolis-Hastings algorithm and we derive some results needed for the Gibbs sampler scheme implemented to sample from the graph posterior distribution $p(\mathcal{G} \mid \mathbb{X})$ in *Section 3.3*.

3.1 Gaussian decomposable graphical model

As mentioned, we first assume that Gaussian data are observed. Specifically, we consider a decomposable graph $\mathcal{G} = (V, E) \in \mathcal{S}_q$ and a Gaussian random vector $\mathbf{Z} = (Z_1, \dots Z_q) \mid \Sigma, \mathcal{G} \sim \mathcal{N}_q(\underline{0}, \Sigma)$, where importantly the (unknown) covariance matrix Σ depends on graph $\mathcal{G}$ and hence satisfies its Markov property.

In particular, under the Gaussian assumption and the undirected graph $\mathcal{G}$, we have that Z_i and Z_j $(i \neq j)$ are conditionally independent given the remaining variables if and only if $\Sigma^{-1}_{i,j} = 0$. Accordingly,

$$\Sigma^{-1}_{i,j} = 0 \Leftrightarrow Z_i \perp\!\!\!\perp Z_j \mid \mathbf{Z}_{V \setminus \{i,j\}} \Rightarrow (i,j) \notin E \tag{23}$$

where $V \setminus \{i, j\}$ denotes the set of nodes in the graph excluding nodes i and j.

Still in this framework, we let $\underline{z}^{(i)} = (z^{(i)}_1, \dots, z^{(i)}_q) \in \mathbb{R}^q$ be the i-th realization of the random vector $\mathbf{Z}$, for $i = 1, \dots n$. Again, we can exploit the *factorization property* of $\mathcal{G}$ decomposable to write the joint density of $\underline{z}^{(i)}$ as

$$p(\underline{z}^{(i)} \mid \Sigma, \mathcal{G}) = \frac{\prod_{C \in \mathcal{C}} p(\underline{z}^{(i)}_C \mid \Sigma_C, \mathcal{G})}{\prod_{S \in \mathcal{S}} p(\underline{z}^{(i)}_S \mid \Sigma_S, \mathcal{G})} \tag{24}$$

where $\underline{z}^{(i)}_C$ denotes the sub-vector of $\underline{z}^{(i)}$ with components indexed by C, while $\Sigma_C = (\Sigma_{ij})_{i,j \in C}$ represents the sub-matrix of Σ obtained selecting rows and columns indexed by C.

In what follows, we present all the ingredients needed to define a graphical model for Gaussian random variables, *i.e* the joint density and the prior on graph $\mathcal{G}$ and the graph-dependent parameter Σ.

3.1.1 Joint density

Given n independent realizations of $\mathbf{Z}$ the likelihood function can be written as

$$\begin{aligned} p(\underline{z}^{(1)}, \dots \underline{z}^{(n)} \mid \Sigma, \mathcal{G}) = \prod_{i=1}^{n} p(\underline{z}^{(i)} \mid \Sigma, \mathcal{G}) &= \frac{1}{(2\pi)^{nq/2} |\Sigma|^{n/2}} exp\left\{ -\frac{1}{2} \sum_{i=1}^{n} \underline{z}^{(i)^T} \Sigma^{-1} \underline{z}^{(i)} \right\} \\ &\propto \frac{\prod_{C \in \mathcal{C}} |\Sigma_C|^{-n/2} exp\left\{ -\frac{1}{2} \sum_{i=1}^{n} \underline{z}^{(i)^T}_C \Sigma^{-1}_C \underline{z}^{(i)}_C \right\}}{\prod_{S \in \mathcal{S}} |\Sigma_S|^{-n/2} exp\left\{ -\frac{1}{2} \sum_{i=1}^{n} \underline{z}^{(i)^T}_S \Sigma^{-1}_S \underline{z}^{(i)}_S \right\}}. \end{aligned} \tag{25}$$

where the last equation results from the factorization in (24).

We emphasize that expression (25) is not the likelihood function of our final model, since what we actually observe are categorical data generated from latent Gaussian variables.

3.1.2 Prior on graph-dependent parameter

First of all, we assume that also in this case the *factorization property* holds, so that the conditional prior on Σ can be written as

$$p(\Sigma \mid \mathcal{G}) = \frac{\prod_{C \in \mathcal{C}} p(\Sigma_C \mid \mathcal{G})}{\prod_{S \in \mathcal{S}} p(\Sigma_S \mid \mathcal{G})}. \tag{26}$$

For prior specification, we still resort to a conjugate prior proposed by Dawid and Lauritzen (1993), namely the *Hyper-Inverse-Wishart* prior; see also [3].

Therefore we will write $\Sigma \mid \mathcal{G} \sim HIW(b, D)$, with $b \in \mathbb{R}^+$ *degrees of freedom* and $D \in M^+(\mathcal{G})$, where $M^+(\mathcal{G})$ denotes the set of all symmetric positive definite matrices having $\Sigma^{-1}_{ij} = 0 \ \forall (i,j) \notin E$ (E is the set of edges of graph $\mathcal{G}$). In particular, $\forall C \in \mathcal{C}$ (and in an analogous way $\forall S \in \mathcal{S}$), the prior is defined as

$$p(\Sigma_C \mid \mathcal{G}) \propto |\Sigma_C|^{-(\frac{b}{2}+|C|)} exp\left\{-\frac{1}{2} tr(\Sigma_C^{-1} D_C)\right\}. \tag{27}$$

We will write $\Sigma_C \mid \mathcal{G} \sim Inv-Wish(b, D_C)$, where as usual D_C is the sub-matrix of D indexed by $C \in \mathcal{C}$.

We remark that Equation (26) holds if for each $S = C_1 \cap C_2$, the elements of Σ_S are in common between Σ_{C_1} and Σ_{C_2}; see also [3].

3.1.3 Prior on the graph

Decomposable graph $\mathcal{G}$ belongs to $\mathcal{S}_q$, the space of all decomposable graphs with q nodes. We refer to *Section 2.2.3* for prior specifications over $\mathcal{S}_q$.

3.1.4 Model specification

Finally, the *Normal-Inverse-Wishart model* can be defined as follows

$$\begin{aligned} \mathcal{Z} \mid \Sigma, \mathcal{G} \sim p(\underline{z}^{(1)}, \dots \underline{z}^{(n)} \mid \Sigma, \mathcal{G}) &\propto \frac{\prod_{C \in \mathcal{C}} |\Sigma_C|^{-n/2} exp\left\{-\frac{1}{2} \sum_{i=1}^n \underline{z}_C^{(i)^T} \Sigma_C^{-1} \underline{z}_C^{(i)}\right\}}{\prod_{S \in \mathcal{S}} |\Sigma_S|^{-n/2} exp\left\{-\frac{1}{2} \sum_{i=1}^n \underline{z}_S^{(i)^T} \Sigma_S^{-1} \underline{z}_S^{(i)}\right\}}; \\ \Sigma \mid \mathcal{G} &\sim HIW(b, D); \\ \mathcal{G} &\sim p(\mathcal{G}). \end{aligned} \tag{28}$$

3.1.5 Posterior on the graph-dependent parameter

Given the previous model specification, we now proceed by deriving the posterior distribution of Σ. In particular, because of conjugacy of the Hyper-Inverse-Wishart prior, the posterior distribution is still Hyper-Inverse-Wishart. For a given clique $C \in \mathcal{C}$, we can express the *posterior density* of θ_C as

$$\begin{aligned} p(\Sigma_C \mid \underline{z}_C^{(1)}, \dots \underline{z}_C^{(n)}, \mathcal{G}) &\propto p(\Sigma_C \mid \mathcal{G}) p(\underline{z}_C^{(1)}, \dots \underline{z}_C^{(n)} \mid \Sigma_C, \mathcal{G}) \\ &= |\Sigma_C|^{-(\frac{b}{2}+|C|)} exp\left\{-\frac{1}{2} tr(\Sigma_C^{-1} D_C)\right\} \prod_{i=1}^n p(\underline{z}_C^{(i)} \mid \Sigma_C, \mathcal{G}) \\ &= |\Sigma_C|^{-(\frac{b}{2}+|C|)} exp\left\{-\frac{1}{2} tr(\Sigma_C^{-1} D_C)\right\} |\Sigma_C|^{-n/2} exp\left\{-\frac{1}{2} \sum_{i=1}^n \underline{z}_C^{(i)^T} \Sigma_C^{-1} \underline{z}_C^{(i)}\right\} \end{aligned} \tag{29}$$

where the last passage is obtained by exploiting (25) and (27). Since the trace operator $tr(\cdot)$ is linear, through simple calculations we obtain the following expression for (29):

$$p(\Sigma_C \mid \underline{z}_C^{(1)}, \dots \underline{z}_C^{(n)}, \mathcal{G}) \propto |\Sigma_C|^{-(\frac{b+n}{2}+|C|)} exp\left\{-\frac{1}{2} tr\left(\Sigma_C^{-1}\left[D_C + \sum_{i=1}^n \underline{z}_C^{(i)} \underline{z}_C^{(i)^T}\right]\right)\right\} \tag{30}$$

from which it is straightforward to derive that

$$\Sigma_C \mid \underline{z}_C^{(1)}, \dots \underline{z}_C^{(n)}, \mathcal{G} \sim Inv - Wish\left(b+n, D_C + \sum_{i=1}^{n} \underline{z}_C^{(i)} \underline{z}_C^{(i)T}\right). \tag{31}$$

A similar procedure working on a fixed separator $S \in \mathcal{S}$ leads to

$$\Sigma_S \mid \underline{z}_S^{(1)}, \dots \underline{z}_S^{(n)}, \mathcal{G} \sim Inv - Wish\left(b+n, D_S + \sum_{i=1}^{n} \underline{z}_S^{(i)} \underline{z}_S^{(i)T}\right) \tag{32}$$

so that, in general, the posterior density of Σ can simply be written as

$$\Sigma \mid \underline{z}^{(1)}, \dots \underline{z}^{(n)}, \mathcal{G} \sim HIW\left(b+n, D + \sum_{i=1}^{n} \underline{z}^{(i)} \underline{z}^{(i)T}\right). \tag{33}$$

3.1.6 Marginal likelihood

The posterior probability of graph $\mathcal{G}$ given the data $\underline{z}^{(1)}, \dots \underline{z}^{(n)}$ is given by

$$p(\mathcal{G} \mid \underline{z}^{(1)}, \dots \underline{z}^{(n)}) \propto m(\underline{z}^{(1)}, \dots \underline{z}^{(n)} \mid \mathcal{G}) p(\mathcal{G})$$

where the *marginal likelihood* $m(\underline{z}^{(1)}, \dots \underline{z}^{(n)} \mid \mathcal{G})$ is again available in closed form. In particular, it can be written as

$$\begin{aligned} & m(\underline{z}^{(1)}, \dots \underline{z}^{(n)} \mid \mathcal{G}) = (2\pi)^{-nq/2} \frac{h(\mathcal{G}, b, D)}{h(\mathcal{G}, b^*, D^*)}; \\ & b^* = b + n; \\ & D^* = D + \sum_{i=1}^{n} \underline{z}^{(i)} \underline{z}^{(i)^T} \end{aligned} \tag{34}$$

where the normalizing constant $h(\cdot)$ is given by

$$h(\mathcal{G}, b, D) = \frac{\prod_{C \in \mathcal{C}} |\frac{1}{2} D_C|^{\frac{b+|C|-1}{2}} \Gamma_{|C|}\left(\frac{b+|C|-1}{2}\right)^{-1}}{\prod_{S \in \mathcal{S}} |\frac{1}{2} D_S|^{\frac{b+|S|-1}{2}} \Gamma_{|S|}\left(\frac{b+|S|-1}{2}\right)^{-1}} \tag{35}$$

and $\Gamma_p(\cdot)$ denotes the *multivariate gamma function*, defined as

$$\Gamma_p(x) = \pi^{\frac{p(p-1)}{4}} \prod_{j=1}^{p} \Gamma\left(x + \frac{1-j}{2}\right). \tag{36}$$

3.2 Categorical variables with latent Gaussian approach

Given the Gaussian (latent) model presented before, we now introduce the model formulation for the collection of categorical (binary) random variables. Specifically, we assume that binary data are generated through *discretization* of their latent Gaussian counterpart, and write, for each $X_j \in \{0, 1\}$, $j = 1, \dots q$,

$$X_j = \begin{cases} 0 & \text{if } Z_j < \theta_0^{(j)} \\ 1 & \text{if } Z_j \geq \theta_0^{(j)} \end{cases} \tag{37}$$

where Z_j is the (latent) Gaussian random variable as defined in *Section 3.1*, while $\theta_0^{(j)}$ represents an unknown cut-off parameter, on which we will also assign a suitable prior.

Observed data will consist of $\underline{x}^{(1)},\dots \underline{x}^{(n)}$ with $\underline{x}^{(i)}$ denoting the i-th realization of the random vector $\mathbf{X} = (X_1,\dots X_q)$.

We detail in the following the main components of the model, and in particular the likelihood function and the prior for parameters $\Theta = \left(\theta_0^{(1)},\dots,\theta_0^{(q)}\right)$. Notice that in this setting we impose a graphical structure on the parameter of the latent vector $\mathbf{Z}$, Σ, for which a suitable prior has been assigned in *Section 3.1*.

3.2.1 Augmented density

We first write the *augmented* density of the observed and latent variables, $(X_1,\dots,X_q,Z_1,\dots,Z_q)$, as

$$p(\underline{x},\underline{z} \mid \Sigma,\Theta) = p(\underline{z} \mid \Sigma)p(\underline{x} \mid \underline{z},\Theta) = p(\underline{z} \mid \Sigma)\mathbb{1}\{\underline{z} \in C(\underline{x},\Theta)\} \tag{38}$$

where $C(\underline{x},\Theta)$ is defined as

$$C(\underline{x},\Theta) = \left[\theta_{x_1-1}^{(1)},\theta_{x_1}^{(1)}\right) \times \left[\theta_{x_2-1}^{(2)},\theta_{x_2}^{(2)}\right) \times \cdots \times \left[\theta_{x_q-1}^{(q)},\theta_{x_q}^{(q)}\right) \tag{39}$$

and we adopt for convenience the notation $\theta_{-1}^{(j)} = -\infty$, $\theta_1^{(j)} = +\infty$, while $\theta_0^{(j)} \in (-\infty,+\infty)$ represents the unknown cut-off for each $j = 1,\dots,q$.

Notice that for example,

$$x_1 = 0 \Rightarrow \left[\theta_{x_1-1}^{(1)},\theta_{x_1}^{(1)}\right) = \left[\theta_{-1}^{(1)},\theta_0^{(1)}\right) = \left(-\infty,\theta_0^{(1)}\right);$$

$$x_1 = 1 \Rightarrow \left[\theta_{x_1-1}^{(1)},\theta_{x_1}^{(1)}\right) = \left[\theta_0^{(1)},\theta_1^{(1)}\right) = \left(\theta_0^{(1)},+\infty\right)$$

coherently with (37).

Based on the model assumption for the latent variables introduced in Section 3.2, we can write explicitly the augmented density as

$$p(\underline{x},\underline{z} \mid \Sigma,\Theta) \propto |\Sigma|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\underline{z}^T\Sigma^{-1}\underline{z}\right)\mathbb{1}(\underline{z} \in C(\underline{x},\Theta)) = \frac{\prod_{C\in\mathcal{C}}|\Sigma_C|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\underline{z}_C^T\Sigma_C^{-1}\underline{z}_C\right)}{\prod_{S\in\mathcal{S}}|\Sigma_S|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}\underline{z}_S^T\Sigma_S^{-1}\underline{z}_S\right)}\mathbb{1}(\underline{z} \in C(\underline{x},\Theta)). \tag{40}$$

3.2.2 Augmented likelihood

Consider now an $n \times q$ *data matrix* $\mathbb{X}$ and the corresponding *latent data matrix* $\mathbb{Z}$ with same dimensions. Then, the augmented likelihood can be written as

$$p(\mathbb{X},\mathbb{Z} \mid \Sigma,\Theta,\mathcal{G}) = \prod_{i=1}^{n} p(\underline{x}^{(i)},\underline{z}^{(i)} \mid \Sigma,\Theta) = \prod_{i=1}^{n} p(\underline{z}^{(i)} \mid \Sigma)\mathbb{1}\left\{\underline{z}^{(i)} \in C(\underline{x}^{(i)},\Theta)\right\} = p(\mathbb{Z} \mid \Sigma)\prod_{i=1}^{n}\mathbb{1}\left\{\underline{z}^{(i)} \in C(\underline{x}^{(i)},\Theta)\right\}. \tag{41}$$

Exploiting the factorization property on $p(\mathbb{Z} \mid \Sigma)$ we obtain:

$$p(\mathbb{X},\mathbb{Z} \mid \Sigma,\Theta,\mathcal{G}) \propto \frac{\prod_{C\in\mathcal{C}}|\Sigma_C|^{-\frac{n}{2}}\exp\left(-\frac{1}{2}tr(\Sigma_C^{-1}S_C)\right)}{\prod_{S\in\mathcal{S}}|\Sigma_S|^{-\frac{n}{2}}\exp\left(-\frac{1}{2}tr(\Sigma_S^{-1}S_S)\right)}\prod_{i=1}^{n}\mathbb{1}\left\{\underline{z}^{(i)} \in C(\underline{x}^{(i)},\Theta)\right\} \tag{42}$$

with $S = \mathbb{Z}^T\mathbb{Z} = \sum_{i=1}^{n}\underline{z}^{(i)}\underline{z}^{(i)^T}$.

3.2.3 Priors

As mentioned in the previous section, parameters $\theta_0^{(j)}$, $j = 1, \dots, q$, are random thresholds linking the latent data with the binary observations.

We assign independent Normal priors with zero mean and variance τ^2 to the q thresholds, namely:

$$\theta_0^{(j)} \overset{iid}{\sim} \mathcal{N}(0, \tau^2).$$

The prior for the graph-dependent parameter Σ is instead assigned as in *Section 3.1.2*:

$$\Sigma \mid \mathcal{G} \sim HIW(b, D).$$

Finally, the prior specification for the graph $\mathcal{G}$ has already been discussed in *Section* 2.2.3.

3.2.4 Posterior distribution

The posterior distribution for our model is given by

$$p(\Sigma, \Theta, \mathcal{G}, \mathbb{Z} \mid \mathbb{X}) = p(\mathbb{X}, \mathbb{Z} \mid \Sigma, \Theta, \mathcal{G}) p(\Sigma \mid \mathcal{G}) p(\mathcal{G}) \prod_{j=1}^{q} p(\theta_0^{(j)}). \tag{43}$$

Notice that we also include among the parameters the latent data matrix, since the latter is not observed; accordingly, we will also need to sample from the full conditional distribution of the latent observations as detailed below.

Recalling the expression of the augmented likelihood $p(\mathbb{X}, \mathbb{Z} \mid \Sigma, \Theta, \mathcal{G})$ in (42), we can write the posterior as

$$p(\Sigma, \Theta, \mathcal{G}, \mathbb{Z} \mid \mathbb{X}) = p(\mathbb{Z} \mid \Sigma) \prod_{i=1}^{n} \mathbb{1}\left\{\underline{z}^{(i)} \in C(\underline{x}^{(i)}, \Theta)\right\} p(\Sigma \mid \mathcal{G}) p(\mathcal{G}) \prod_{j=1}^{q} p(\theta_0^{(j)}) \tag{44}$$

where

$$p(\mathbb{Z} \mid \Sigma) = \prod_{i=1}^{n} p(\underline{z}^{(i)} \mid \Sigma). \tag{45}$$

by assuming independence among the n realizations of $\mathbf{Z}$ given Σ and $\mathcal{G}$.

3.3 Metropolis-Hastings algorithm

In this section we describe the Metropolis-Hastings procedure implemented to sample from the graph posterior distribution. Notice that

$$p(\mathcal{G} \mid \mathbb{X}) \propto p(\Sigma, \Theta, \mathcal{G}, \mathbb{Z} \mid \mathbb{X}).$$

However, in this case, sampling directly from $p(\Sigma, \Theta, \mathcal{G}, \mathbb{Z} \mid \mathbb{X})$ is not possible. Therefore, we need to implement a Gibbs sampling scheme together with a Metropolis-Hastings step.

3.3.1 Gibbs sampler: full conditionals

First of all, we need to derive the *full conditional distributions* for all the parameters of interest.

We could in principle write the full conditional of each single parameter given all the others, but in the following we will choose suitable blocks of parameters for convenience. In particular we will consider:

1. The full conditional of $(\Sigma, \mathcal{G})$ given the rest, *i.e.* $p(\Sigma, \mathcal{G} \mid \Theta, \mathbb{Z}, \mathbb{X})$;
2. The full conditional of $(\mathbb{Z}, \Theta)$ given the rest, *i.e.* $p(\mathbb{Z}, \Theta \mid \Sigma, \mathcal{G}, \mathbb{X})$.

Starting from the full conditional of $(\Sigma, \mathcal{G})$, we have

$$p(\Sigma, \mathcal{G} \mid \Theta, \mathbb{Z}, \mathbb{X}) \propto p(\mathbb{Z} \mid \Sigma) p(\Sigma \mid \mathcal{G}) p(\mathcal{G}) \tag{46}$$

where we notice that $p(\Sigma, \mathcal{G} \mid \Theta, \mathbb{Z}, \mathbb{X})$ reduces to $p(\Sigma, \mathcal{G} \mid \mathbb{Z})$ since there is no dependence on $\mathbb{X}$ and Θ once $\mathbb{Z}$ is known. In particular, the three terms on the right hand side of (46) have already been presented in the *Normal-Inverse-Wishart* model specified in *Section 3.1* and it is not difficult to sample from them.

With regard to the full conditional of $(\mathbb{Z}, \Theta)$, we write it for convenience as

$$p(\mathbb{Z}, \Theta \mid \Sigma, \mathcal{G}, \mathbb{X}) = p(\mathbb{Z} \mid \Theta, \Sigma, \mathcal{G}, \mathbb{X}) p(\Theta \mid \Sigma, \mathcal{G}, \mathbb{X}). \tag{47}$$

In particular, the first term can be written as

$$\begin{aligned} p(\mathbb{Z} \mid \Theta, \Sigma, \mathcal{G}, \mathbb{X}) &\propto p(\mathbb{Z} \mid \Sigma) \prod_{i=1}^{n} \mathbb{1}\left\{ \underline{z}^{(i)} \in C(\underline{x}^{(i)}, \Theta) \right\} \\ &= \prod_{i=1}^{n} p(\underline{z}^{(i)} \mid \Sigma) \mathbb{1}\left\{ \underline{z}^{(i)} \in C(\underline{x}^{(i)}, \Theta) \right\} \\ &= \prod_{i=1}^{n} d\mathcal{N}_q(\underline{z}^{(i)} \mid \underline{0}, \Sigma) \mathbb{1}\left\{ \underline{z}^{(i)} \in C(\underline{x}^{(i)}, \Theta) \right\} \end{aligned} \tag{48}$$

so that we can sample each $\underline{z}^{(i)}$ for $i = 1, \dots n$ independently from a suitable *Multivariate Truncated Normal* distribution.

The second term instead can be expressed as

$$\begin{aligned} p(\Theta \mid \Sigma, \mathcal{G}, \mathbb{X}) = p(\theta_0^{(1)}, \dots \theta_0^{(q)} \mid \Sigma, \mathcal{G}, \mathbb{X}) &= \int_{\mathbb{R}^{n \times q}} p(\Theta, \mathbb{Z} \mid \Sigma, \mathcal{G}, \mathbb{X}) \, d\mathbb{Z} \\ &\propto \int_{\mathbb{R}^{n \times q}} \left\{ \prod_{i=1}^{n} p(\underline{z}^{(i)} \mid \Sigma) \mathbb{1}\left\{ \underline{z}^{(i)} \in C(\underline{x}^{(i)}, \Theta) \right\} d\underline{z}^{(1)} \dots d\underline{z}^{(n)} \right\} \cdot \prod_{j=1}^{q} p(\theta_0^{(j)}) \\ &= \prod_{i=1}^{n} \int_{\mathbb{R}^q} p(\underline{z}^{(i)} \mid \Sigma) \mathbb{1}\left\{ \underline{z}^{(i)} \in C(\underline{x}^{(i)}, \Theta) \right\} d\underline{z}^{(i)} \cdot \prod_{j=1}^{q} p(\theta_0^{(j)}) \\ &= \prod_{i=1}^{n} \left\{ \Phi_q(\theta^1_{x_{ij}}, \dots \theta^q_{x_{ij}} \mid \Sigma) - \Phi_q(\theta^1_{x_{ij}-1}, \dots \theta^q_{x_{ij}-1} \mid \Sigma) \right\} \cdot \prod_{j=1}^{q} p(\theta_0^{(j)}). \end{aligned} \tag{49}$$

In particular, for $j = 1, \dots, q$, we update $\theta_0^{(j)}$ sequentially using a Metropolis-Hastings scheme based on the following steps:

- propose $(\theta_0^{(j)})^*$ from the proposal distribution $q\left((\theta_0^{(j)})^* \mid \theta_0^{(j)} \right) \sim \mathcal{N}\left(\theta_0^{(j)}, \sigma_0^2 \right)$;
- given the current $\theta_0^{(j)}$ accept $(\theta_0^{(j)})^*$ with probability

$$\alpha_j = \min\left\{ 1 \, ; \, \frac{p\left((\theta_0^{(j)})^* \, , \, \underline{\theta}_0^{-j} \mid \Sigma, \mathcal{G}, \mathbb{X} \right)}{p\left(\theta_0^{(j)} \, , \, \underline{\theta}_0^{-j} \mid \Sigma, \mathcal{G}, \mathbb{X} \right)} \cdot \frac{q\left(\theta_0^{(j)} \mid (\underline{\theta}_0^{-j})^* \right)}{q\left((\underline{\theta}_0^{-j})^* \mid \theta_0^{(j)} \right)} \right\}$$

where $\underline{\theta}_0^{-j} = \{\theta_0^{(k)}, k \neq j\}$ represents the vector of all cut-offs except the j-th.

3.3.2 Implementation of the algorithm

Once that the preliminary step in the Gibbs sampler has been done, the Metropolis-Hastings procedure works exactly the same as in the *Multinomial-Hyper-Dirichlet* model.

In particular, the proposal distribution $q(\mathcal{G}' \mid \mathcal{G})$ is obtained as described in *Section 2.3.1*, building $\mathcal{G}'$ by either adding or removing an edge from $\mathcal{G}$.

For convenience, in the following we report the full implementation of the Metropolis-Hastings procedure to sample from the posterior distribution $p(\mathcal{G} \mid \underline{x}^{(1)}, \ldots, \underline{x}^{(n)})$.

Algorithm MH algorithm for the Latent-Normal-Inverse-Wishart Model

---

Input: $\mathcal{G}^{(0)}$ (the initial candidate graph), M (the number of MCMC iterations), $\Theta^{(0)}$ (the initial cut-offs vector), $\mathbb{Z}^{(0)}$ (the initial latent Gaussian data), $\Sigma^{(0)}$ (the initial covariance matrix)

Output: An MCMC sample $\{\mathcal{G}^{(t)}\}_{t=1}^{M}$ from the graph posterior distribution

for $t \leftarrow 1$ **to** M **do**

$\Theta^{(t)} \leftarrow$ Metropolis-Hastings step $\left(\Sigma^{(t-1)}, \mathbb{X}, \tau^2, \Theta^{(t-1)}\right)$;

$\Sigma^{-1^{(t)}} \sim HIW\left(b+n,\ D + \mathbb{Z}^{(t-1)^T}\mathbb{Z}^{(t-1)},\ \mathcal{G}^{(t-1)}\right)$;

$\mathbb{Z}^{(t)} \sim t\mathcal{N}\left(\Sigma^{(t)}, \Theta^{(t)}, \mathbb{X}\right)$;

$\mathcal{G} \leftarrow \mathcal{G}^{(t-1)}$;

$\mathcal{G}' \sim q(\mathcal{G}' \mid \mathcal{G})$;

compute $\alpha(\mathcal{G}' \mid \mathcal{G}) = \min\left\{1, \frac{m(\mathbb{Z}^{(t)}|\mathcal{G}')}{m(\mathbb{Z}^{(t)}|\mathcal{G})} \cdot \frac{p(\mathcal{G}')}{p(\mathcal{G})}\right\}$;

update $\mathcal{G}^{(t)} = \begin{cases} \mathcal{G}', & \text{with probability } \alpha \\ \mathcal{G}^{(t-1)}, & \text{with probability } 1-\alpha \end{cases}$

end

---

4 Cluster analysis

In this section we consider *mixtures of graphical models* to account for possible heterogeneous dependence relations among subjects, which can be linked to a latent clustering structure of the data. We base our model formulation on a *Dirichlet Process* (DP) mixture of graphical models, that for simplicity we base on the *Multinomial-Hyper-Dirichlet* model described in *Section 2*.

4.1 Dirichlet process mixture models

We first summarize the main features of a *Dirichlet Process Mixture Model*, using the typical notation found in literature; see also [6].

The basic structure applies to data $y_1, \ldots, y_n$ independently drawn from some distribution, where each y_i could also be a multivariate categorical observation. We model the distribution from which each y_i is drawn as a mixture of distributions of the form $F(\theta)$, with the mixing distribution over θ denoted as M. We let the prior for this mixing distribution be a *Dirichlet Process*, with *concentration parameter* α and *base distribution* M_0. The DP mixture model can therefore be written in the following hierarchical structure:

$$
\begin{aligned}
y_i \mid \theta_i &\sim F(\theta_i); \\
\theta_i \mid M &\sim M; \\
M &\sim DP(M_0, \alpha)
\end{aligned}
\tag{50}
$$

where $DP(M_0, \alpha)$ represents the Dirichlet Process with base distribution M_0 and concentration parameter α.

In a DP mixture model each observation y_i can in principle have its own parameter θ_i, however all of them are drawn from the same family of distributions. A clustering structure of the data arises from the presence of *ties* in the collection $\theta_1, \ldots, \theta_n$: if two individuals share the same parameter, then they are assigned to the same cluster. Parameters $\theta_1, \ldots, \theta_n$ are drawn from M, a common prior, which we may call a "distribution over distributions".

4.2 Finite mixture models

An equivalent representation of a DP mixture model can be obtained by taking the limit as K goes to infinity of a finite mixture model with K components (clusters) of the form:

$$
\begin{aligned}
y_i \mid c_i, \boldsymbol{\phi} &\sim F(\phi_{c_i}); \\
\phi_k &\sim M_0; \\
c_i \mid \boldsymbol{p} &\sim Discrete(p_1, ..., p_K); \\
\boldsymbol{p} &\sim Dirichlet(\alpha/K, ..., \alpha/K).
\end{aligned}
\tag{51}
$$

Here, each $c_i \in \{1, ..., K\}$ is a random variable indexing the cluster to which the i-th observation belongs, while $\boldsymbol{\phi} = (\phi_{c_i})_{i=1}^n$ is a collection of K cluster-specific parameters: accordingly, ϕ_{c_i} denotes the c_i-th parameter, meaning the parameter that we associate to subject i as member of class c_i.

A prior on c_i is assigned as a discrete distribution with parameter vector $\boldsymbol{p} = (p_1, ..., p_K)$, where p_k is the probability that *a priori* subject i is assigned to cluster k (and of course it must hold $\sum_{k=1}^K p_k = 1$).

4.3 Inference on clustering

The latter formulation of a DP mixture model allows to easily integrate over the mixing proportions $\boldsymbol{p}$ to obtain the prior for the indicator variable c_i. This can be expressed as the product of conditional probabilities of the following form:

$$
\begin{aligned}
\mathbb{P}(c_i = k \mid c_1, \ldots, c_{i-1}) &= \frac{\mathbb{P}(c_1, \ldots, c_{i-1}, c_i = k)}{\mathbb{P}(c_1, \ldots, c_{i-1})} \\
&= \frac{\int p_{c_1} \ldots p_{c_{i-1}} p_k \Gamma(\alpha) \Gamma(\alpha/K)^{-K} p_1^{(\alpha/K)-1} \ldots p_K^{(\alpha/K)-1} d\mathbf{p}}{\int p_{c_1} \ldots p_{c_{i-1}} \Gamma(\alpha) \Gamma(\alpha/K)^{-K} p_1^{(\alpha/K)-1} \ldots p_K^{(\alpha/K)-1} d\mathbf{p}} \\
&= \frac{n_{i,k} + \alpha/K}{i - 1 + \alpha}
\end{aligned}
\tag{52}
$$

where $n_{i,k}$ is the number of c_j for $j < i$ that are equal to k (see also [9] for further details).

If we now let K go to infinity, the conditional probabilities which define the prior for the indicator c_i reach the following limits:

$$\begin{aligned}
&\mathbb{P}(c_i = k \mid c_1, \ldots, c_{i-1}) \overset{K\to+\infty}{\longrightarrow} \frac{n_{i,k}}{i-1+\alpha};\\
&\mathbb{P}(c_i \neq c_j, \forall j < i \mid c_1, \ldots, c_{i-1}) \overset{K\to+\infty}{\longrightarrow} \frac{\alpha}{i-1+\alpha}.
\end{aligned} \tag{53}$$

The interpretation of the two limits is straightforward: the first one states that the probability that the i-th observation belongs to cluster k is proportional to the number of past observations that belong to that cluster, meaning that the larger the k-th cluster is, the more likely the i-th observation will belong to that cluster.

The second one, instead, states that there is a non-null probability that subject i will be assigned to a new cluster (i.e. different from the others) and this probability is proportional to α. Therefore, the role of α is that of a "precision" parameter as it controls the variability of the distribution over M: in other words, larger values of α will encourage the allocation of a given subject in a new cluster.

The hyperparameter α can be fixed based on some prior knowledge on the expected number of clusters; alternatively, a prior on α can be assigned through a Gamma distribution; see for instance [5] for full details.

4.4 Algorithm

To sample from the posterior of the DP mixture model we follow *Algorithm 2* in [9]. The latter applies to conjugate models whereas the model-dependent parameter can be integrated out.

In our *Multinomial-Hyper-Dirichlet* setting we can integrate out parameter $\underline{\theta}$ to obtain the marginal likelihood of the graph (see *Section 2.2)*. However, we do not integrate out $\mathcal{G}$, as we are interested in both clustering and graph estimation.

Given an initial state where the data $y_1, ..., y_n$ have been divided into K clusters through indicator variables $c_1, \ldots, c_n$ and graphs $\mathcal{G}_1, \ldots, \mathcal{G}_K$ are associated with each of the components, the algorithm proceeds to sample from the joint conditional distribution of $\left(K, \{c_i\}_{i=1}^n, \{\mathcal{G}_k\}_{k=1}^K, \alpha \;\middle|\; y_1, ..., y_n\right)$.

4.4.1 First step

As a first stage we update the sequence of indicators $\{c_i\}_{i=1}^n$ and, implicitly, the number of clusters K using a Gibbs sampling scheme by sequentially sampling each c_i for $i = 1, \ldots, n$ from its full conditional distribution, integrating over θ_k. In particular:

$$\begin{aligned}
&\text{If } c_i = c_j \text{ for some } j \neq i\text{: } \mathbb{P}(c_i = k \mid c_{-i}, y_i, \mathcal{G}) = b\frac{n_{-i,k}}{n-1+\alpha}\int F(y_i \mid \underline{\theta}, \mathcal{G}_k) dH_{-i,k}(\underline{\theta} \mid \mathcal{G}_k);\\
&\text{If } c_i \neq c_j \;\forall j \neq i\text{: } \mathbb{P}(c_i \neq c_j \,\forall j \neq i \mid c_{-i}, y_i, \mathcal{G}) = b\frac{\alpha}{n-1+\alpha}\int F(y_i \mid \underline{\theta}, \mathcal{G}^*) dM_0(\underline{\theta} \mid \mathcal{G}^*).
\end{aligned} \tag{54}$$

where the symbols denote the following:

- $n_{-i,k}$ is the number of c_j, $j \neq i$ that are equal to k;
- b is an appropriate normalizing constant;
- $H_{-i,k}$ is the posterior distribution of $\underline{\theta}$ based on the prior M_0, all the observations y_j for which $j \neq i$, $c_i = k$ and the graph $\mathcal{G}_k$;

- $\mathcal{G}^*$ is an empty cluster (graph) which has to be randomly sampled from our baseline measure on $\mathcal{S}_q$.

Since the two integrals are predictive distributions and since we consider a conjugate model, we are able to calculate them in closed form. Thus, we can update our parameter c_i, for $i = 1,\dots,n$, drawing a new value from $(c_i \mid c_{-i}, y_i, \mathcal{G})$ as defined in (54).

Notice that if the last remaining observation has been removed from a cluster, that cluster is deleted and K is decreased by 1; similarly, if an observation is moved to a new cluster that is currently empty K is increased by 1.

4.4.2 Second step

Once the cluster assignment has been updated, each graph $\mathcal{G}_k$ associated with cluster k, for $k = 1,...,K$, is also updated. We first define a partition of the observations based on the cluster to which they belong at step t in the following way:

$$C_k := \{y_i : c_i = k\}, \ \forall k = 1,\dots,K.$$

Accordingly, we can compute the marginal likelihood of graph $\mathcal{G}_k$, namely $m(\mathcal{G}_k \mid C_k)$ for each $k = 1,\dots,K$.

Update of graph $\mathcal{G}_k$ is performed through a Metropolis-Hastings step (exactly as in the MCMC scheme of *Section 2.3.2*) Specifically, we first propose a new graph $\mathcal{G}'_k$ from the proposal distribution $q(\mathcal{G}'_k \mid \mathcal{G}_k)$ (see also *Section 2.3.1*)and accept the new graph with probability:

$$\alpha_k = \min\left\{1, \frac{m(\mathcal{G}'_k \mid C_k)}{m(\mathcal{G}_k \mid C_k)} \cdot \frac{p(\mathcal{G}'_k)}{p(\mathcal{G}_k)}\right\}.$$

5 Results

5.1 Assessment of the performances

All the algorithms presented in the previous sections have been implemented using the statistical programming language R. An assessment of the performances of the two models for inference on the graph, namely the Multinomial-Dirichlet and the Latent Normal Inverse-Wishart models, has been obtained via the following methodology. We first created 20 categorical datasets starting from as many randomly generated decomposable graphs (with 6 nodes) and then we ran both algorithms on each of these datasets. After that, we computed the *Structural Hamming Distance* between the original graph generating the data and the graph estimated from the resulting chain. In particular, two ways of estimating the graph have been considered:

1. *Median Probability Graph*: the graph obtained by including all the edges that were in at least 50% of the graphs visited by the chain;

2. *Maximum a Posteriori*: the most recurrent graph in the chain.

The results obtained are shown in Figure 1 (Multinomial-Dirichlet model) and Figure 2 (Latent Normal Inverse-Wishart Model).

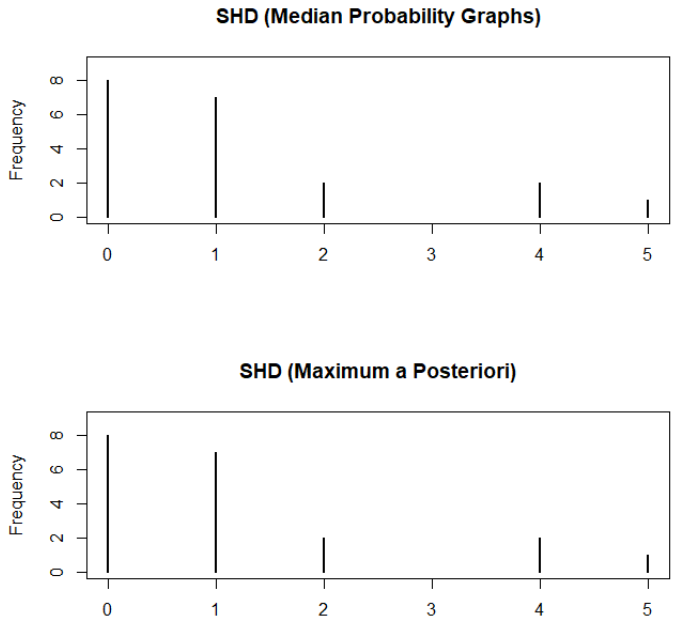

Figure 1: Results for the Multinomial-Dirichlet Model.

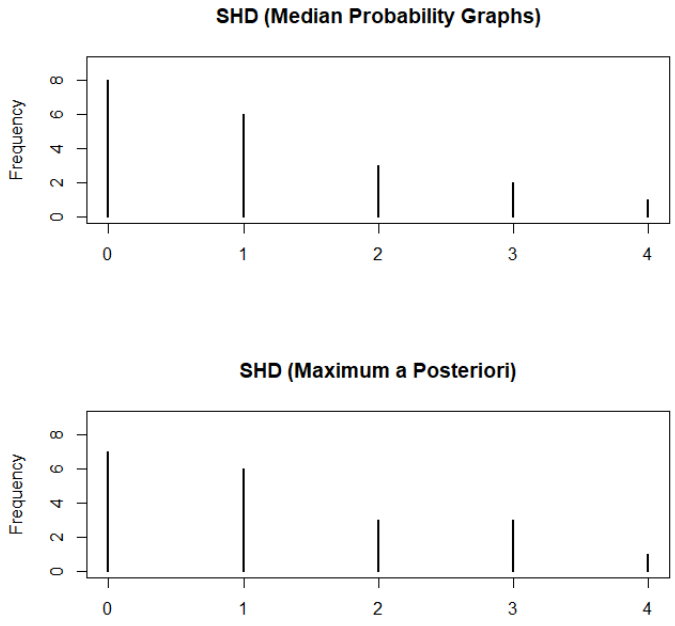

Figure 2: Results for the Latent Normal Inverse-Wishart Model.

As we can see from the plots, both algorithms have good performances, with a Structural Hamming Distance between the true graph and the estimated one smaller than 2 in most of the cases. However, we shall notice that the Multinomial-Dirichlet model has better performance for what concerns the inference results as well as the computational cost.

5.2 Inference on the Congressional Voting Records dataset

So far, we have applied the described models only on simulated data. We now perform inference on the *Congressional Voting Records* dataset. Such dataset includes votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the CQA, namely:

1. Handicapped Infants;
2. Water Project Cost Sharing;
3. Adoption of the Budget Resolution;
4. Physician Fee Freeze;
5. El Salvador Aid;
6. Religious Groups in Schools;
7. Anti Satellite Test Ban;
8. Aid to Nicaraguan Contras;
9. Mx Missile;
10. Immigration;
11. Synfuels Corporation Cutback;
12. Education Spending;
13. Superfund Right to Sue;
14. Crime;
15. Export Administration Act South Africa;
16. Duty Free Exports.

In particular, we divided the congressmen into 2 groups (republicans and democrats) and we ran the Multinomial-Dirichlet model individually on each of them. Note that we chose the Multimonial-Dirichlet model mainly because of the lower computational time. The results obtained are shown in Figure 3 (republicans) and Figure 4 (democrats).

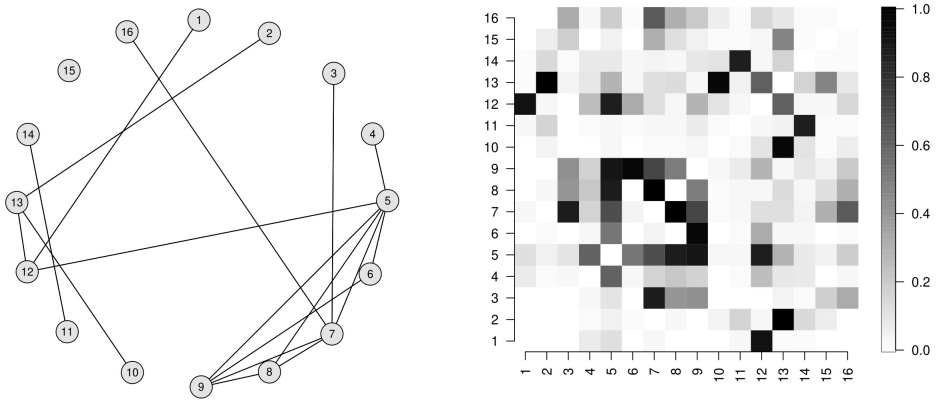

Figure 3: Results for the Republicans: on the left the estimated Median Probability Graph and on the right the heatmap of the edge-inclusion probabilities.

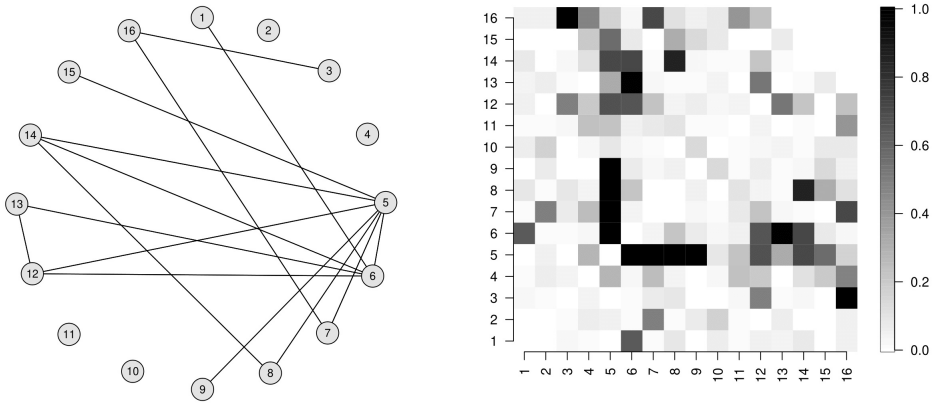

Figure 4: Results for the Democrats: on the left the estimated Median Probability Graph and on the right the heatmap of the edge-inclusion probabilities.

As for possible interpretations of the results, let us first focus on the isolated nodes. Such nodes represent the questions whose answers are independent from the answer to any other question. In particular, we can notice that in the graph in Figure 3 (which is referred to the republican congressmen) the only isolated node is the node 15, whereas in the graph in Figure 4 (which is referred to the democrat congressmen) we have 4 isolated nodes, namely 2, 4, 10 and 11. The fact that such sets of nodes differ among the two graphs may suggest that the votes of a congressman are influenced by their party. On the other hand, we can further observe that both graphs are characterized by some common structure. For instance, the nodes 5 and 6 are highly central (in terms of the number of connections) in both cases. In particular, these sets of (strongly dependent) nodes may reflect a firm position of the two parties on the subjects corresponding to such questions.

6 Future developments

Some possible future developments of the project are the following:

1. Improve the computational efficiency of the Metropolis-Hastings algorithm implementing the Latent Normal Inverse-Wishart model. In particular, we observed that the bottleneck of the performances is given by the sampling from the multivariate truncated Gaussian distributions;

2. Extend the Latent Normal Inverse-Wishart model to the case of non-binary ordinal variables;

3. Implement the clustering algorithm using a Dirichlet Process mixture of Latent Normal Inverse-Wishart models. Indeed, so far we have only implemented the algorithm using a mixture of Multinomial-Dirichlet models;

4. Improve the efficiency of the clustering algorithm built upon the mixture of Multinomial-Dirichlet models. In particular, we have already considered many possible improvements in order to avoid repeated computations. However, the results obtained are not satisfying yet and it may be necessary to implement the algorithm using some faster programming language (such as C++).

References

[1] Davide Altomare, Guido Consonni, and Luca La Rocca. Objective bayesian search of gaussian directed acyclic graphical models for ordered variables with non-local priors. *Biometrics*, 69 2:478–87, 2013.

[2] Carlos M. Carvalho and James G. Scott. Objective bayesian model selection in gaussian graphical models. *Biometrika*, 96:497–512, 2009.

[3] A. Philip Dawid and Steffen L. Lauritzen. Hyper markov laws in the statistical analysis of decomposable graphical models. *Annals of Statistics*, 21:1272–1317, 1993.

[4] Adrian Dobra and Alex Lenkoski. Copula gaussian graphical models and their application to modeling functional disability data. *The Annals of Applied Statistics*, 5:969–993, 2011.

[5] Michael D. Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[6] Thomas S. Ferguson. A bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1:209–230, 1973.

[7] Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Graphical models for ordinal data. *Journal of Computational and Graphical Statistics*, 24:183 – 204, 2015.

[8] Marloes H. Maathuis, Mathias Drton, Steffen L. Lauritzen, and Martin J. Wainwright. Handbook of graphical models. 2018.

[9] Radford M. Neal. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249 – 265, 2000.

[10] Abel Rodríguez, Alex Lenkoski, and Adrian Dobra. Sparse covariance estimation in heterogeneous samples. *Electronic journal of statistics*, 5:981–1014, 2011.

[11] Hao Wang and Sophia Zhengzi Li. Efficient gaussian graphical model determination under g-wishart prior distributions. *Electronic Journal of Statistics*, 6:168–198, 2012.