École Polytechnique Fédérale de Lausanne

Bayesian joint inference of multiple graphical models using spike-and-slab priors

by Luca Bracone

Master Thesis

Approved by the Examining Committee:

Hélène Ruffieux
Thesis Advisor

Expert Reviewer (who?)
External Expert

Anthony Davison
Thesis Supervisor

EPFL MA STAT
Station 14
CH-1015 Lausanne

January 10, 2023

Abstract

Graphical models are models in which the conditional independence of variables is represented as a graph. If the variables are assumed to be distributed following a multivariate Gaussian distribution, conditional independence between pairs of variables is equivalent to the corresponding entries of the precision matrix bieng zero. Using a spike-and-slab prior we aim to model the zero and non-zero entries as a mixture. Rather than using Gibbs-sampling methods, our work makes use of an expectation conditional maximisation algorithm (ECM) in order to obtain fast pointwise estimates. We extend previously done work by focusing on the analysis of multiple graphs. In doing so we leverage shared information across graphs to obtain better estimates. We show on simulated data that our method produces better estimates than other single-graph methods.

Chapter 1

Introduction

In recent times, there has been an increased interest in finding complex relationships underlying biological processes, such as gene expression pathways or connections between neurons in the brain. In the past, many approaches have focused on *directed graphical models*, in which nodes are random variables and the structure of edges forces the joint distribution to factor in a certain way. Some approaches have instead focused on *undirected graphical models*, in which the nodes are also some variables of interest, but in which the existence of edges imposes a certain conditional independence structure on the variables. This report develops methods to infer edges in an undirected graph.

We use a Bayesian framework. It allows us to specify a prior distribution over the graphs, which can encode specific domain knowledge. For instance, the estimated graph is often chosen to be sparse, i.e, to have few edges. There are many possible priors one can choose from. The conjugate G-Wishart prior derived by Roverato (2002) has been a common choice. Recently, other priors have been used because they proved to have better computational scalability as the number of parameters increases. Those include graphical horseshoe (Y. Li, Craig, and Bhadra, 2019), the spike-and-slab graphical lasso (Z. R. Li, McCormick, and Clark, 2018), and the spike-and-slab (Wang, 2015) which is the one we will use.

Most inference approaches for undirected Bayesian graphs have focused on stochastic methods which obtain an estimate of the full posterior distribution using numerical sampling methods such as Markov chain Monte Carlo (MCMC) with Gibbs sampling. However, for many practical applications point estimates are sufficient, so we will follow Z. R. Li and McCormick (2017), who derive an expectation conditional maximisation (ECM) approach to inference. We will then extend their method to multiple graphs. Lukemire et al. (2017) have a method that is fairly similar to ours, but we will use a probit link to pool information across the graphs (they use a logistic link), and we will also infer pairwise similarity between the graphs. In Chapter 2 we will cover the basics of Bayesian hierarchical and graphical models. In Chapter 3 we will derive the multi-graph model, and an ECM algorithm to fit it. In Chapter 4 we will highilight some methods to choose hyperparameters. Finally, In Chapter 5 we will perform simulations and assess the performance of the multi-graph model.

Chapter 2

Undirected graphical models for multivariate Gaussian variables

2.1 Bayesian hierarchical models

A hierarchical model involves conditional prior distributions over all model parameters $\theta_1, \ldots, \theta_p$. This results in a hierarchical structure, and the higher a parameter is placed up the hierarchy, the higher the number of samples we need to have to produce confident estimates of it. For example, suppose that we observe values $y_{ij} \overset{\text{i.i.d.}}{\sim} P(\theta_j)$ where $j = 1, \ldots, p$ and $i = 1, \ldots, n_j$. That is, we have a certain number of observations, each belonging to some group, and the values within each group have parameter θ_j. We will now discuss two methods to analyze such data that will motivate the use of hierarchical models.

It might at first seem appealing to ignore the differences between groups. This means setting all the θ_j to be equal to each other, so we can perform usual maximum likelihood estimation. If there is a group with only a few outlying observations, the maximum likelihood estimator will have high bias, and it will have an estimate that is far from its observations. Then another idea might be to treat each group in a separate estimation procedure, assuming that they are unrelated. In some cases, this is justified, but the group with only a few observations will have an estimate with a large variance.

A hierarchical model sees the first two methods as two extremes: “the θ_j are the same” and “the θ_j are unrelated”. Would there be a way to automatically decide how similar or how different the θ_j should be from each other? Yes, we will imagine that the θ_j are themselves independent and identically distributed $\theta_j \mid \phi \sim Q(\phi)$, for some parameter ϕ. Then the posterior joint distribution can be expressed as

$$p(\phi, \theta_1, \ldots, \theta_p \mid y) \propto p(y \mid \theta) p(\theta_1, \ldots, \theta_p \mid \phi) p(\phi).$$

The distribution $p(\phi)$ chosen for ϕ, is often referred to as a *hyperprior*.

2.2 Undirected graphical models

An undirected graphical model is a model in which the conditional structure of some variables of interest $y_1, \ldots, y_p$ are represented using a graph $G = (V, E)$, with $V = \{y_1, \ldots, y_p\}$ such that an edge (y_i, y_j) exists in E if and only if y_i and y_j are dependent given all the other variables (which we denote as $y_{-(ij)}$, for $i, j \in \{1, \ldots, p\}$). So in summary

$$y_i \not\perp\!\!\!\perp y_j \mid y_{-(ij)} \iff (y_i, y_j) \in E.$$

For a sample of n p-dimensional multivariate Gaussian variables $y^1, \ldots, y^n \sim \mathrm{N}_p(0, \Sigma)$ we would like to deduce the structure of the graph G by estimating the *precision matrix* $\Omega = \Sigma^{-1}$ and by observing that a given entry ω_{ij} is zero if and only if y_i and y_j are independent given $y_{-(ij)}$, i.e.,

$$(y_i, y_j) \notin E \iff \omega_{ij} = 0.$$

2.3 Spike and slab prior for graphical models

Let $y \in \mathbb{R}^p$ be random vector distributed under the hierarchical model

$$\begin{aligned} y \mid \Omega &\sim \mathrm{N}_p(0, \Omega^{-1}), \quad \Omega \in M^+, \\ \omega_{ij} \mid \delta_{ij} &\sim \delta_{ij}\mathrm{N}(0, v_1^2) + (1 - \delta_{ij})\mathrm{N}(0, v_0^2), \quad v_0^2 \ll v_1^2, \quad i, j = 1, \ldots, p, \quad i \neq j, \\ \omega_{ii} &\sim \textsc{Exp}(\lambda/2), \\ \delta_{ij} \mid \pi &\sim \textsc{Bern}(\pi), \\ \pi &\sim \textsc{Beta}(a, b), \end{aligned}$$

where M^+ is the set of symmetric positive definite matrices, $\mathrm{N}_p(0, \Omega^{-1})$ is the multivariate normal distribution with mean 0 and covariance matrix Ω^{-1}, and $a, b, \lambda, v_0^2, v_1^2 \in \mathbb{R}^+$ are hyperparameters. The entries ω_{ij} are so that the conditional distribution of Ω as a whole can be written as

$$p(\Omega \mid \delta) = C^{-1} \prod_{i<j} \mathrm{N}(\omega_{ij} \mid 0, v_{\delta_{ij}}^2) \prod_i \textsc{Exp}\left(\omega_{ii} \mid \frac{\lambda}{2}\right) \mathbb{1}\{\Omega \in M^+\},$$

with C a constant that depends on $\delta, v_0^2, v_1^2, \lambda$. This is known as the continuous *spike-and-slab* prior because it corresponds to a mixture of two Gaussian distributions, one with a small variance v_0^2 (the spike) and one with a large variance v_1^2 (the slab). Under this continuous spike-and-slab, if an entry ω_{ij} is truly zero, it is absorbed in the spike and estimated as close to zero. The discrete spike-and-slab instead uses a point mass at zero, $\mathbb{1}\{\omega_{ij} = 0\}$. We use the former because it makes the computation of the posterior distribution simpler. Finally, let $Y \in \mathbb{R}^{n \times p}$ be the matrix whose rows are identically and independently distributed observations of y. We seek values of Ω, δ, π that maximise the log posterior joint distribution

$\log p(\Omega, \delta, \pi \mid Y)$. The posterior joint distribution can be decomposed as

$$p(\Omega, \delta, \pi \mid Y) = p(\Omega \mid \delta) p(\delta \mid \pi) p(Y \mid \Omega) p(\pi) p(Y)^{-1}. \tag{2.1}$$

The factor $p(Y)^{-1}$ is constant, and hence has no influence on the maximisation. Using the definitions and the decomposition of (2.1), we find that $\log p(\Omega, \delta, \pi \mid Y)$ equals

$$\begin{aligned} \text{constant} + \sum_{i<j} \left[-\frac{1}{2} \log \left\{ v_0^2 (1-\delta_{ij}) + v_1^2 \delta_{ij} \right\} - \frac{\omega_{ij}^2}{2} \frac{1}{v_0^2(1-\delta_{ij}) + v_1^2 \delta_{ij}} \right] - \sum_i \frac{\lambda}{2} \omega_{ii} \\ + \sum_{i<j} \left\{ \delta_{ij} \log(\pi) + (1-\delta_{ij}) \log(1-\pi) \right\} \\ + (a-1)\log(\pi) + (b-1)\log(1-\pi) + \frac{n}{2} \log\det(\Omega) - \frac{1}{2} \operatorname{tr}(Y^t Y \Omega). \end{aligned} \tag{2.2}$$

2.4 Expectation Maximisation for Bayesian graphical models

Following Z. R. Li and McCormick (2017) instead of maximising (2.2), we iteratively maximise its expectation over δ, which we implement using an iterative "expectation conditional maximization" (ECM) approach. Taking the expectation of (2.2) we obtain

$$Q(\Omega, \pi \mid \Omega^{(l)}, \pi^{(l)}, Y) = \mathbb{E}_{\delta \mid \Omega^{(l)}, \pi^{(l)}, Y} \left\{ \log p(\Omega, \delta, \pi | X) \;\middle|\; \Omega^{(l)}, \pi^{(l)}, Y \right\}, \tag{2.3}$$

where $\Omega^{(l)}$ and $\pi^{(l)}$ denote the values obtained for Ω and π at the l-th iteration of the algorithm, respectively. Equation (2.3) is equal to

$$\begin{aligned} \text{constant} - \sum_{i<j} \frac{\omega_{ij}^2}{2} \mathbb{E}_{\delta_{ij}|\cdot} \left(\frac{1}{v_0^2(1-\delta_{ij}) + v_1^2 \delta_{ij}} \right) - \sum_i \frac{\lambda}{2} \omega_{ii} \\ + \frac{p(p-1)}{2} \log(1-\pi) + \sum_{i<j} \mathbb{E}_{\delta_{ij}|\cdot}(\delta_{ij}) \log\left(\frac{\pi}{1-\pi} \right) \\ + (a-1)\log(\pi) + (b-1)\log(1-\pi) + \frac{n}{2} \log\det(\Omega) - \frac{1}{2} \operatorname{tr}(Y^t Y \Omega), \end{aligned} \tag{2.4}$$

where $\mathbb{E}_{\delta_{ij}|\cdot}$ denotes the conditional expectation with respect to $\delta_{ij} \mid \Omega^{(l)}, \pi^{(l)}, Y$. The expectation terms are

$$\mathbb{E}_{\delta_{ij}|\cdot}(\delta_{ij}) = p\left(\delta_{ij} = 1 \mid \omega_{ij}^{(l)}, \pi^{(l)}\right) = \frac{\pi^{(l)} p\left(\omega_{ij}^{(l)} \mid \delta_{ij} = 1\right)}{\pi^{(l)} p\left(\omega_{ij}^{(l)} \mid \delta_{ij} = 1\right) + \left(1 - \pi^{(l)}\right) p\left(\omega_{ij}^{(l)} \mid \delta_{ij} = 0\right)}, \tag{2.5}$$

which we denote q_{ij}, and

$$d_{ij} := \mathbb{E}_{\delta_{ij}|\cdot}\left(\frac{1}{v_0^2(1-\delta_{ij})+v_1^2\delta_{ij}}\right) = \sum_{\delta=0}^{1}\frac{p\left(\delta \mid \omega_{ij}^{(l)}, \pi^{(l)}\right)}{v_0^2(1-\delta)+v_1^2\delta} = \frac{q_{ij}}{v_1^2} + \frac{1-q_{ij}}{v_0^2}. \tag{2.6}$$

This is the *expectation step* (E step). Now we use (2.5) and (2.6) to compute the next iterates $\pi^{(l+1)}$ and $\Omega^{(l+1)}$. The derivative of (2.3) with respect to π is

$$\pi\left\{\frac{p(p-1)}{2} - a - b + 2\right\} + \sum_{i<j} q_{ij} + a - 1,$$

and it is equal to zero when

$$\pi = \frac{a - 1 + \sum_{i<j} q_{ij}}{a + b - 2 + \frac{p(p-1)}{2}}.$$

The maximisation with respect to Ω requires that Ω remains positive definite after each iteration. In the context of Gibbs sampling, Wang (2015) has shown that if we slice the matrices Ω, Y^tY and $V = \left(v_{\delta_{ij}}\right)_{ij}$ in the following way

$$\Omega = \begin{pmatrix}\Omega_{11} & \omega_{12}\\ \omega_{12}^t & \omega_{22}\end{pmatrix}, \quad Y^tY = \begin{pmatrix}S_{11} & s_{12}\\ s_{12}^t & s_{22}\end{pmatrix}, \quad V = \begin{pmatrix}V_{11} & v_{12}\\ v_{12}^t & v_{22}\end{pmatrix},$$

where ω_{22} is a scalar and ω_{12} is a $(p-1)$-dimensional vector (likewise for s_{22}, s_{12}, and v_{12}, v_{22}), we find the conditional distributions

$$\omega_{12} \mid \delta, Y \sim \mathrm{N}(-Cs_{12}, C) \quad C = \left\{(s_{22}+\lambda)\Omega_{11}^{-1} + \mathrm{diag}\left(v_{12}^{-1}\right)\right\}^{-1},$$

and

$$\omega_{22} \mid \delta, Y \sim \textsc{Gamma}\left(\frac{n}{2}+1, \frac{s_{22}+\lambda}{2}\right) + \omega_{12}^t\Omega_{11}^{-1}\omega_{12}.$$

The term v_{12}^{-1} refers to the vector v_{12} after we inverted each component, so $\mathbb{E}\left(v_{12}^{-1}\right) = d_{12}$, where d_{12} is the vector of d_{ij} values defined similarly as ω_{12}. Taking the mode of these distributions gives

$$\begin{aligned}\omega_{12}^{(l+1)} &= -\left\{(s_{22}+\lambda)\Omega_{11}^{-1} + \mathrm{diag}(d_{12})\right\}^{-1}s_{12},\\ \omega_{22}^{(l+1)} &= \frac{n}{s_{22}+\lambda} + \left(\omega_{12}^{(l+1)}\right)^t\Omega_{11}^{-1}\omega_{12}^{(l+1)},\end{aligned}$$

which results in a *conditional maximisation step* (CM-step). Inference is performed by alternating between the E and CM steps, until convergence of Q, for some prespecified tolerance.

Chapter 3

Extending to multiple graphs

In some applications, it is reasonable to assume that we have two or more different graphs (for instance, diseased vs. healthy patients). So now, we suppose that each sample belongs to one of K groups, where K is the number of graphs. This means that we have K sets $\{y_k^1, \ldots, y_k^{n_k}\}$, each of which we assume to be realizations of a multivariate gaussian distribution

$$y_k \mid \Omega_k \sim \mathrm{N}_p(0, \Omega_k^{-1}), \quad \Omega_k \in M^+,$$

where M^+ is the set of positive-definite matrices. We would like to make use of a hierarchical model to pool information across $\Omega_1, \ldots, \Omega_K$. We will use the following model

$$\begin{aligned} y_k \mid \Omega_k &\sim \mathrm{N}_p(0, \Omega_k^{-1}), \quad \Omega_k \in M^+, \quad k = 1, \ldots, K \\ \omega_{ijk} \mid \delta_{ijk} &\sim \delta_{ijk}\mathrm{N}(0, v_{1,k}^2) + (1 - \delta_{ijk})\mathrm{N}(0, v_{0,k}^2), \quad v_{0,k}^2 \ll v_{1,k}^2, \quad i, j = 1, \ldots, p, \quad i \neq j \\ \omega_{iik} &\sim \textsc{Exp}(\lambda_k/2), \\ \delta_{ijk} \mid \theta_{ijk} &\sim \textsc{Bern}(\Phi(\theta_{ijk})) \\ \theta_{ij} &\sim \mathrm{N}_K(\theta_0, \Sigma_0), \end{aligned} \tag{3.1}$$

where Φ is the standard normal cumulative distribution function, $v_{0,k}^2, v_{1,k}^2, \lambda_k, \Sigma \in \mathbb{R}^+$ are hyperparameters, and $\theta_0 < 0$ is a hyperparameter to induce sparsity. In Chapter 4 we discuss how they are chosen. We use the following data augmentation for inference

$$\delta_{ijk} = \mathbb{1}\left\{z_{ijk} > 0\right\}, \quad \text{where } z_{ijk} \mid \theta_{ijk} \sim N\left(\theta_{ijk}, 1\right).$$

Let Y_k be the matrix whose rows are observations of y_k, and let Y be $\{Y_1, \ldots, Y_K\}$. The posterior joint distribution $p(\Omega, z, \theta \mid Y)$ decomposes as

$$\begin{aligned} p(\Omega, z, \theta \mid Y) &= p(Y \mid \Omega)p(\Omega \mid z)p(z \mid \theta)p(\theta)p(Y)^{-1} \\ &= \prod_{k=1}^{K} p(Y_k \mid \Omega_k) \prod_{i<j} \prod_{k=1}^{K} p(\omega_{ijk} \mid z_{ijk}) \prod_{i<j} \prod_{k=1}^{K} p(z_{ijk} \mid \theta_{ijk}) \prod_{i<j} p(\theta_{ij}) p(Y)^{-1}. \end{aligned} \tag{3.2}$$

When we take the log of (3.2) and unravel the formula we obtain

$$
\begin{aligned}
&-\frac{Kp(p-1)}{2}\log(2\pi)-\frac{p(p-1)}{4}\log\det(\Sigma)+\sum_{k=1}^{K}\frac{n_k}{2}\log\det(\Omega_k)-\sum_{k=1}^{K}\frac{pn_k}{2}\log(2\pi)\\
&-\sum_{k=1}^{K}\frac{1}{2}\operatorname{tr}(S_k\Omega_k)+\sum_{i<j}\sum_{k=1}^{K}-\frac{1}{2}\log(2\pi v_{\delta_{ijk},k})-\sum_{i<j}\sum_{k=1}^{K}\frac{\omega_{ijk}^2}{2v_{\delta_{ijk},k}^2}+\sum_{k=1}^{K}p\log\left(\frac{\lambda_k}{2}\right)-\sum_{k=1}^{K}\frac{\lambda_k}{2}\operatorname{tr}(\Omega_k)\\
&-\frac{1}{2}\sum_{i<j}\sum_{k=1}^{K}(z_{ijk}-\theta_{ijk})^2-\frac{1}{2}\sum_{i<j}\theta_{ij}^t\Sigma^{-1}\theta_{ij}-\log p(Y)+\epsilon(\theta_0) \quad (3.3)
\end{aligned}
$$

where $S_k = Y_k^tY_k$, and $\epsilon(\theta_0)$ are other terms involving θ_0. When we take the conditional expectation of (3.3) over the latent variable, z_k, we obtain

$$
\begin{aligned}
&Q(\Omega,\theta\mid\Omega^{(l)},\theta^{(l)},Y)=\mathbb{E}_{z\mid\Omega^{(l)},\theta^{(l)},Y}\left\{\log p(\Omega,z,\theta\mid Y)\mid\Omega^{(l)},\theta^{(l)},Y\right\}\\
&=\sum_{k=1}^{K}\frac{n_k}{2}\log\det(\Omega_k)-\frac{1}{2}\sum_{k=1}^{K}\operatorname{tr}(S_k\Omega_k)-\frac{1}{2}\sum_{i<j}\sum_{k=1}^{K}\omega_{ijk}^2\mathbb{E}_{\cdot|\cdot}\left\{\frac{1}{\mathbb{1}(z_{ijk}>0)v_{1,k}^2+\mathbb{1}(z_{ijk}\le 0)v_{0,k}^2}\right\}\\
&-\frac{1}{2}\sum_{k=1}^{K}\lambda_k\operatorname{tr}(\Omega_k)-\frac{1}{2}\sum_{i<j}\sum_{k=1}^{K}\theta_{ijk}^2+\sum_{i<j}\sum_{k=1}^{K}\theta_{ijk}\mathbb{E}_{\cdot|\cdot}(z_{ijk})-\frac{1}{2}\sum_{i<j}\theta_{ij}^t\Sigma_0^{-1}\theta_{ij}+\text{constant} \quad (3.4)
\end{aligned}
$$

where $\mathbb{E}_{z_{ijk}|\cdot}$ refers to the expectation of z_{ijk} conditioned on $\Omega^{(l)},\theta^{(l)}$, and Y. We now proceed to the computation of the expectation terms. First we note that $p(\delta_{ijk}=1\mid Y,\Omega^{(l)},\theta^{(l)})$ is equal to

$$
\begin{aligned}
&\frac{p\left(\omega_{ijk}^{(l)}\mid\delta_{ijk}=1\right)p\left(\delta_{ijk}=1\mid\theta_{ijk}^{(l)}\right)}{p\left(\omega_{ijk}^{(l)}\mid\delta_{ijk}=0\right)p\left(\delta_{ijk}=0\mid\theta_{ijk}^{(l)}\right)+p\left(\omega_{ijk}^{(l)}\mid\delta_{ijk}=1\right)p\left(\delta_{ijk}=1\mid\theta_{ijk}^{(l)}\right)}\\
&=\frac{\mathrm{N}\left(\omega_{ijk}^{(l)}\mid 0,v_{1,k}^2\right)\Phi\left(\theta_{ijk}\right)}{\mathrm{N}\left(\omega_{ijk}^{(l)}\mid 0,v_{0,k}^2\right)\left\{1-\Phi\left(\theta_{ijk}\right)\right\}+\mathrm{N}\left(\omega_{ijk}^{(l)}\mid 0,v_{1,k}^2\right)\Phi\left(\theta_{ijk}\right)}
\end{aligned}
$$

Then the first expectation term is given by

$$
\begin{aligned}
&\mathbb{E}_{\cdot|\cdot}\left\{\frac{1}{v_{0,k}^2\mathbb{1}(z_{ijk}\le 0)+v_{1,k}^2\mathbb{1}(z_{ijk}>0)}\right\}\\
&=\mathbb{E}_{\cdot|\cdot}\left\{\frac{1}{v_{0,k}^2(1-\delta_{ijk})+v_{1,k}^2\delta_{ijk}}\right\}\\
&=\frac{p\left(\delta_{ijk}=1\mid Y,\Omega^{(l)},\theta^{(l)}\right)}{v_{1,k}^2}+\frac{1-p\left(\delta_{ijk}=1\mid Y,\Omega^{(l)},\theta^{(l)}\right)}{v_{0,k}^2}.
\end{aligned}
$$

To calculate the other expectation term, $\mathbb{E}_{z_{ijk}|\cdot}(z_{ijk})$, we first see that

$$\begin{aligned}
p\left(z_{ijk} \mid \mathbf{Y}, \mathbf{\Omega}^{(l)}, \delta_{ijk}, \theta^{(l)}\right) &= \frac{p\left(z_{ijk}, Y, \Omega^{(l)}, \delta_{ijk}, \theta^{(l)}\right)}{p\left(Y, \Omega^{(l)}, \delta_{ijk}, \theta^{(l)}\right)} \\
&= \frac{p\left(Y \mid \Omega^{(l)}, z_{ijk}, \delta_{ijk}, \theta^{(l)}\right) p\left(\Omega^{(l)} \mid z_{ijk}, \delta_{ijk}, \theta^{(l)}\right) p\left(z_{ijk} \mid \delta_{ijk}, \theta^{(l)}\right) p\left(\delta_{ijk} \mid \theta^{(l)}\right) p\left(\theta^{(l)}\right)}{p\left(Y \mid \Omega^{(l)}, \delta_{ijk}, \theta^{(l)}\right) p\left(\Omega^{(l)} \mid \delta_{ijk}, \theta^{(l)}\right) p\left(\delta_{ijk} \mid \theta^{(l)}\right) p\left(\theta^{(l)}\right)} \\
&= \frac{p\left(Y \mid \Omega^{(l)}\right) p\left(\Omega^{(l)} \mid z_{ijk}\right) p\left(z_{ijk} \mid \delta_{ijk}\right) p\left(\delta_{ijk} \mid \theta^{(l)}\right) p\left(\theta^{(l)}\right)}{p\left(Y \mid \Omega^{(l)}\right) p\left(\Omega^{(l)} \mid \delta_{ijk}\right) p\left(\delta_{ijk} \mid \theta^{(l)}\right) p\left(\theta^{(l)}\right)} \\
&= p(z_{ijk} \mid \delta_{ijk}).
\end{aligned}$$

If $\delta_{ijk} = 1$ then $z_{ijk} \mid \delta_{ijk}$ is the same as $z_{ijk} \mid z_{ijk} > 0$ which is a truncated normal random variable with mean

$$\theta_{ijk}^{(l)} + \frac{\phi(\theta_{ijk}^{(l)})}{\Phi(\theta_{ijk}^{(l)})},$$

where ϕ denotes the PDF of a standard normal random variable. On the other hand, if $\delta_{ijk} = 0$ then $z_{ijk} \mid \delta_{ijk}$ is the same as $z_{ijk} \mid z_{ijk} \leq 0$ which is also a truncated normal random variable with mean

$$\theta_{ijk}^{(l)} - \frac{\phi(\theta_{ijk}^{(l)})}{1 - \Phi(\theta_{ijk}^{(l)})}.$$

Therefore $\mathbb{E}_{z_{ijk}|\cdot}(z_{ijk})$ is equal to

$$\begin{aligned}
&\sum_{\delta_0=0}^{1} \mathbb{E}_{\cdot|\cdot}\left(z_{ijk} \mid \delta_{ijk} = \delta_0, y_k, \Omega_k, \theta_{ijk}\right) p(\delta_{ijk} = \delta_0 \mid y_k, \Omega_k, \theta_{ijk}) \\
&= \left\{\theta_{ijk}^{(l)} + \frac{\phi(\theta_{ijk}^{(l)})}{\Phi(\theta_{ijk}^{(l)})}\right\} p(\delta_{ijk} = 1 \mid \cdot) + \left\{\theta_{ijk}^{(l)} - \frac{\phi(\theta_{ijk}^{(l)})}{1 - \Phi(\theta_{ijk}^{(l)})}\right\} p(\delta_{ijk} = 0 \mid \cdot) \\
&= \theta_{ijk}^{(l)} - \frac{\phi(\theta_{ijk}^{(l)})}{1 - \Phi(\theta_{ijk}^{(l)})} + p(\delta_{ijk} = 1 \mid \cdot) \left\{\frac{\phi(\theta_{ijk}^{(l)})}{\Phi(\theta_{ijk}^{(l)})} + \frac{\phi(\theta_{ijk}^{(l)})}{1 - \Phi(\theta_{ijk}^{(l)})}\right\} \\
&= \theta_{ijk}^{(l)} + M\left(\theta_{ijk}^{(l)}, 0\right) + p(\delta_{ijk} = 1 \mid \cdot) \left\{M\left(\theta_{ijk}^{(l)}, 1\right) - M\left(\theta_{ijk}^{(l)}, 0\right)\right\},
\end{aligned}$$

where $M(\alpha, c)$ denotes Mill's ratio

$$M(\alpha, c) = (-1)^{1-c} \frac{\phi(\alpha)}{\Phi(\alpha)^c \{1 - \Phi(\alpha)\}^{1-c}}.$$

This is the E-step for the multi-graph setting. Now for the M-step, we denote $\Xi = \Sigma_0^{-1}$ and $\xi_{kk'}$ the (k, k')-th entry of Ξ. We differentiate $Q(\Omega, \theta \mid \Omega^{(l)}, \theta^{(l)}, Y)$ with respect to θ_{ijk} to obtain

$$q_{ijk} - \theta_{ijk} - \xi_{kk}\theta_{ijk} - \sum_{\substack{k'=1 \\ k' \neq k}}^{K} \xi_{kk'}\theta_{ijk'}. \tag{3.5}$$

Equation (3.5) is equal to zero when

$$\theta_{ijk} = \frac{q_{ijk} - \sum_{\substack{k'=1 \\ k' \neq k}}^{K} \xi_{kk'}\theta_{ijk'}}{1 + \xi_{kk}}$$

where $q_{ijk} = \mathbb{E}_{\cdot|\cdot}(z_{ijk})$. The updates for Ω are obtained in a similar fashion as before:

$$\begin{aligned} \omega_{k,12}^{(l+1)} &= - \left\{ (s_{k,22} + \lambda_k) \left(\Omega_{k,11}^{(l+1)} \right)^{-1} + \operatorname{diag}(d_{k,12}) \right\}^{-1} s_{k,12}, \\ \omega_{k,22} &= \frac{n_k}{s_{k,22} + \lambda_k} + \left(\omega_{k,12}^{(l+1)} \right)^t \Omega_{k,11}^{-1} \omega_{k,12}^{(l+1)}. \end{aligned}$$

3.1 Adding a prior for Σ_0

Given that Σ_0 is meant to represent how similar pairs of graphs are, we would like such information to be inferred from the data as well, rather than to be imposed by us. We expand the model in (3.1) by specifying the prior

$$\Sigma_0 \sim \mathrm{W}^{-1}(\Psi, \nu),$$

where $\mathrm{W}^{-1}(\Psi, \nu)$ is the *inverse Wishart* distribution whose density is

$$f(\Sigma_0; \Psi, \nu) = \frac{\det(\Psi)^{\frac{\nu}{2}}}{2^{\frac{\nu K}{2}} \Gamma_p(\frac{\nu}{2})} \det(\Sigma_0)^{-\frac{\nu+K+1}{2}} \exp\left\{ -\frac{1}{2} \operatorname{tr}(\Psi \Sigma_0^{-1}) \right\},$$

with parameters Ψ, a positive definite $K \times K$ matrix, and $\nu > K - 1$ a scalar. The Q function in Equation (3.3) is now

$$\begin{aligned} Q\left(\Omega, \theta, \Sigma_0 \mid \Omega^{(l)}, \theta^{(l)}, \Sigma_0^{(l)}, Y\right) &= \mathbb{E}_{z|\Omega^{(l)},\theta^{(l)},\Sigma^{(l)},Y} \left\{ \log p(\Omega, z, \theta, \Sigma_0 \mid Y) \mid \Omega^{(l)}, \theta^{(l)}, \Sigma_0^{(l)}, Y \right\} \\ &= Q\left(\Omega, \theta \mid \Omega^{(l)}, \theta^{(l)}, Y\right) - \frac{2(\nu + K + 1) + p(p-1)}{4} \log\det(\Sigma_0) - \frac{1}{2} \operatorname{tr}(\Psi \Sigma_0^{-1}) + \text{constant}. \end{aligned} \tag{3.6}$$

Adding this new term does not change the computations in the E-step. Let us now compute the posterior distribution of Σ in our model

$$\begin{aligned} p(\Sigma_0 \mid Y, \Omega, z, \theta) &= \frac{p(Y \mid \Omega)p(\Omega \mid z)p(z \mid \theta)p(\theta \mid \Sigma_0)p(\Sigma_0)}{p(Y \mid \Omega)p(\Omega \mid z)p(z \mid \theta)p(\theta)} \\ &= \frac{p(\theta \mid \Sigma_0)p(\Sigma_0)}{p(\theta)} \\ &= p(\Sigma_0 \mid \theta). \end{aligned}$$

Such simplifications are thanks to the fact that Σ_0 appears last in the model. Now, we compute an M-step for Σ_0. We make use of the fact that the inverse Wishart distribution is conjugate to the multivariate Gaussian distribution. That is, if we observe a sample $\{\theta_{ij}\}_{i,j=1,\ldots,p}$ in which each element is $\mathrm{N}_K(\theta_0, \Sigma)$ distributed, then the posterior is

$$\Sigma \mid \Theta \sim \mathrm{W}^{-1}\left(\Psi + \sum_{i<j}(\theta_{ij} - \theta_0)(\theta_{ij} - \theta_0)^t, \frac{p(p-1)}{2} + \nu\right).$$

This motivates the update step in which we simply set $\Sigma^{(l+1)}$ to the mode of the posterior distribution

$$\Sigma^{(l+1)} = \frac{\Psi + \sum_{i<j}(\theta_{ij} - \theta_0)(\theta_{ij} - \theta_0)^t}{\frac{p(p-1)}{2} + \nu + K + 1}.$$

In Chapter 5 we will perform simulations with and without this prior on Σ.

Chapter 4

Choosing hyperparameters

This chapter mainly concerns the choice of v_0 in the single-graph model. The performance of the model is heavily influenced by the values of the hyperparameters. On Figure 4.1 we have plotted the mean AUC for changing values of $v0$ in the single graph setting, fixing $v_1 = 100$. We propose two strategies to choose v_0.

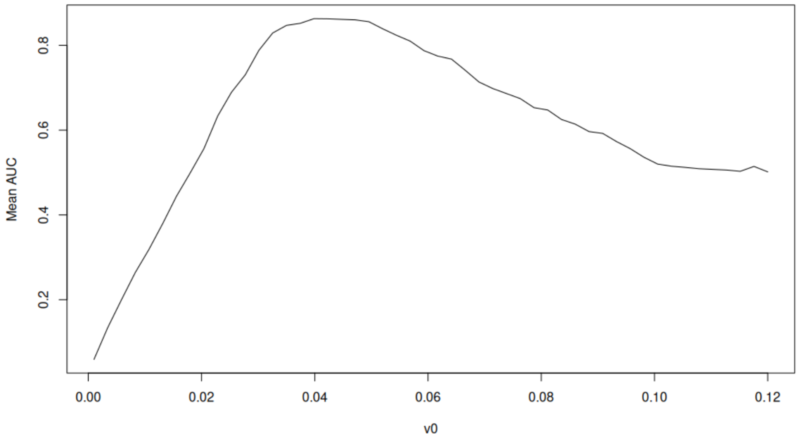

Figure 4.1: Mean AUC over ten replicates for different values of v_0, with a graph with 25 nodes and 100 samples.

4.1 Imposing a given sparsity level

Suppose that the proportion of edges $s \in [0, 1]$ is known. Then, since the parameter π controls the prior sparsity of the graph, we fix $a = 1$ and b so that the mean of π is $a/(a + b) = s$. We then run the single-graph model over a grid of v_0 values and pick the one for which the estimated graph has edge density closest to the s we imposed. On Figure 4.2 we show the result of choosing v_0 with this method. Note that we used the true sparsity for that graph, which was approximately 0.15. It is interesting to see that despite using the true sparsity, we overestimate v_0 with respect to the AUC.

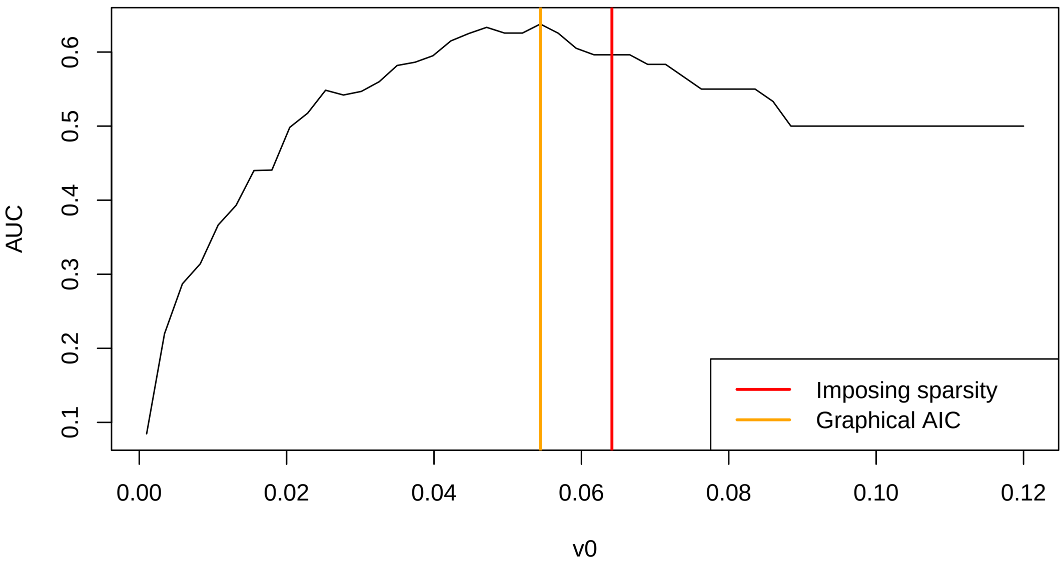

Figure 4.2: Area under curve ("AUC") values for single-graph estimation as a function of v_0, with $n = 50$, $p = 20$. The red line represents the value for v_0 we would choose if we followed the method outlined in Section 4.1, and the orange line the one we would choose with the method in Section 4.2.

4.2 Graphical AIC

The Akaike information criterion (AIC) in the graphical setting is given by the following formula

$$2|E| - \log\det(\Omega) + \operatorname{tr}(S\Omega),$$

where $S = YY^t$ and $Y = [y_1, \ldots, y_n]$. As we can see in Figure 4.3, the value of v_0 that minimizes AIC is the one which maximizes the AUC. Although here we only show the result for one graph, using the AIC in this way generally gives v_0 values that are very close to the best v_0. Other formulas could have been used such as the Bayes information criterion (BIC), or the extended BIC, which tend to be less conservative. For the multiple graphs setting, we run one of the methods described below for each graph individually to obtain multiple values of v_0 with which we fit the mutli-graph model.

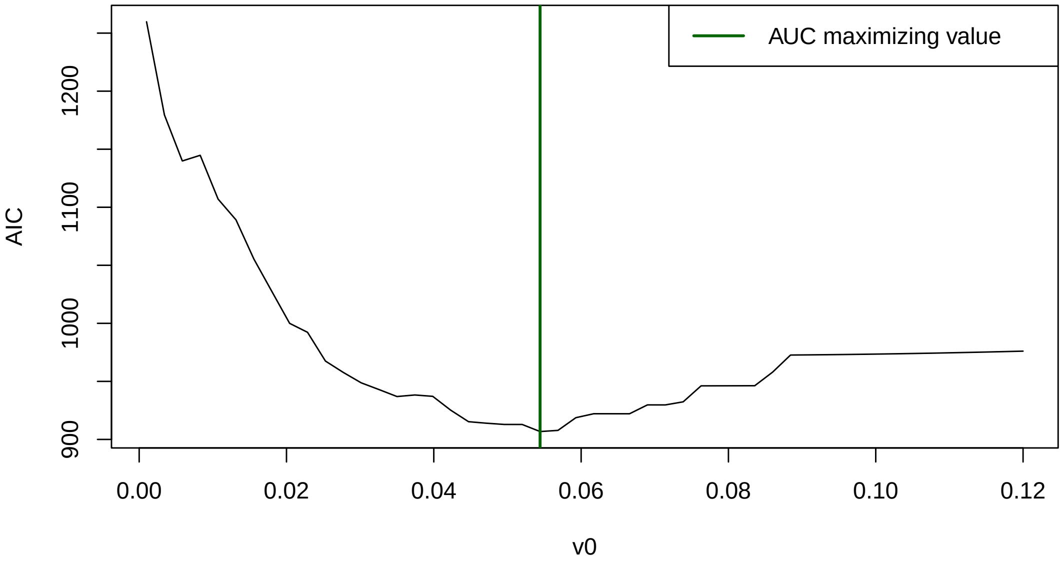

Figure 4.3: AIC values for single-graph estimation as a function of v_0, with $n = 50$, $p = 20$. The green line represents the value of v_0 for which the area under curve (“AUC”) is maximized.

Chapter 5

Simulations

5.1 Performance in the single-graph setting

We performed a set of simulations with data generated from the R package `huge` (Zhao et al., 2020). We compare the results of our method with that of Meinshausen and Buehlmann (2006) which for a given node i, computes

$$\hat{\theta}^{i,\lambda} = \underset{\theta:\theta_i=0}{\operatorname{argmin}} \frac{1}{n}||Y_i - Y\theta||_2^2 + \eta||\theta||_1,$$

where Y_i is the i-th column of Y, $\theta_j^i = -\omega_{ij}/\omega_{ii}$, and η is a constant that controls the l_1 penalty term. We only use one replicate but plan to add more, so we can obtain measures of uncertainty for the performance of our method. The tests were performed on different graph structures that `huge` allows us to generate. In Table 5.1 and 5.2, the lines labelled by “random” indicate that the underlying graph was generated such that each edge had an equal probability of being present. For the lines labelled by “cluster” instead, vertices were split into groups and an edge had a higher probability of appearing when the vertices belonged in the same group.

Our method, EMGS, estimates Ω and π. We use (2.5) to obtain the estimated posterior probabilities of inclusion of all edges (y_i, y_j). The method of Meinshausen and Buehlmann (2006) estimates Ω as a function of η. Higher η values yield a sparser estimate of Ω. We compare maximal *F1 scores* which is the harmonic mean between *precision* and *recall*. The

graph	method	25	50	100
random	EMGS	0.87	0.83	0.75
	mb	0.89	0.85	0.76
cluster	EMGS	0.73	0.70	0.67
	mb	0.75	0.70	0.63

Table 5.1: $n = 50$, $p \in \{25, 50, 100\}$, F1 scores. For EMGS, hyperparameters were selected by maximising F1 score.

graph	method	25	50	100
cluster	EMGS	0.68	0.67	0.67
	mb	0.89	0.87	0.89

Table 5.2: $n = 200$, $p \in \{25, 35, 50\}$. F1 scores. For EMGS, hyperparameters were selected by maximising the posterior joint distribution.

precision is the fraction of true edges among the edges that the method detects, and recall is the fraction of true edges which the method detects. The former is also known as the positive predictive value and the latter as the true positive rate. In this experiment, we selected the "best" v_0 in Table 5.2, and in Table 5.2 we provide a more "honest" approach which picks v_0 by maximising the posterior distribution. However, this often produced degenerate selections with the posterior probabilities of inclusion $p(\delta_{ij} \mid \Omega, \pi, Y)$ collapsing to either zero or one.

5.2 Performance in the multi-graph setting

In this section we show the results of the algorithm derived in Chapter 3.

5.2.1 Data generation

The data for the multi-graph setting is obtained by first taking a graph that was generated with `huge`. Then we make a copy of the original graph, and for each edge we randomly and independently decide if it will be swapped, with some probability. If yes, we move the edge over to a random pair of unconnected vertices. We also swap the relevant entries in the precision matrix. If the resulting precision matrix is not positive definite, we discard it and start over. This process is repeated K times to obtain the desired number of graphs.

5.2.2 Comparison with single-graph methods

We run the algorithm described in Chapter 3 with $n = 50$ and $p = 20$. In Figure 5.1 we show one estimate that the multi-graph method produced. We see that although the true Ω only has zero or positive entries, our method estimated some negative entries. We think that it is due to the symmetric nature of the spike-and-slab prior, but more work is required to understand why such a mistake is occuring. In Figure 5.2 we plot the estimated Ω from the multi-graph model in Chapter 3 against the estimated Ω from the single-graph model in Chapter 2. We see that the multi-graph model improves upon the single-graph model by producing fewer false positives.

We perform a series of experiments. In the first one, we compare our multi-graph model to the single-graph models. The results are summarized in Table 5.3. Although this is expected, we are reassured to see that the multi-graph model performs better than the other single-graph methods (Meinshausen and Buehlmann, 2006, and the single-graph method from Chapter 2). Also, the multi-graph model performs better than its single-graph counterpart even when we

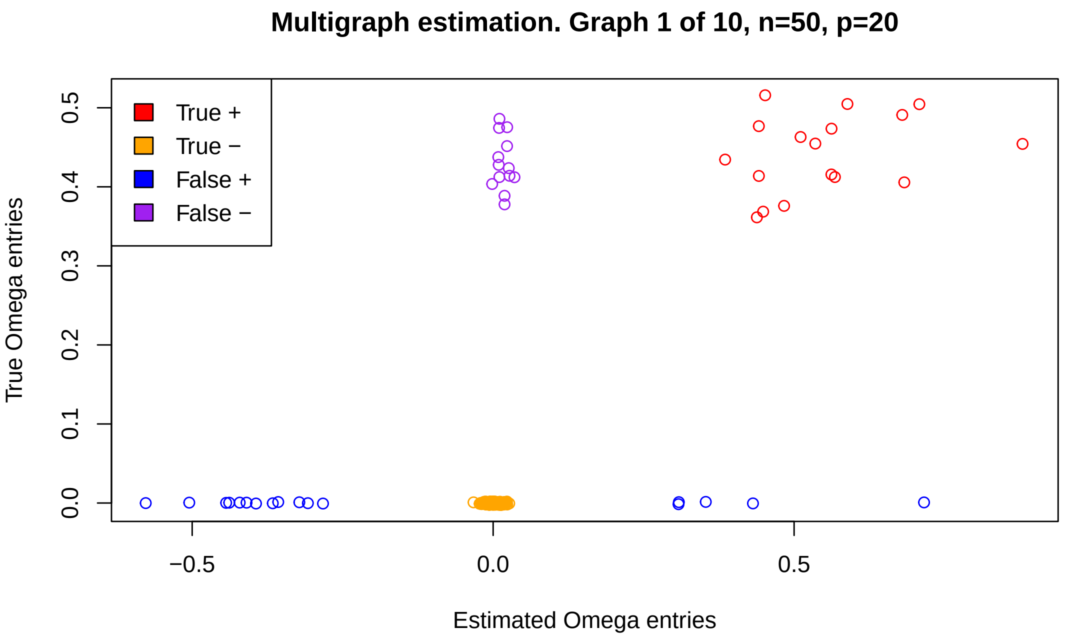

Figure 5.1: The entries of the estimated $\hat{\Omega}_k$ matrix with the multi-graph modlel compared to the true Ω_k matrix, for $k = 1$ in a scatterplot with the multi-graph model. Points in red correspond to true positives, orange are true negatives, blue are false positives, and purple are false negatives. The points have been jittered slightly so that overlapping points are more easily discernable.

only have one graph. Possibly, it seems that choosing a better prior may have had a positive influence on the performance.

5.3 Examining the effect of pooling

In the second experiment we fix $K = 4$ and instead we increase the dissimilarity between graphs. The results are summarized in Table 5.4. Here we only look at the multi-graph model. We see that in general the model performs similarly, regardless of how different the graphs are from eachother. This is due to a bug in the code and will be fixed.

In the third and final experiment, we increase the number of graphs and see the effect on the performance.

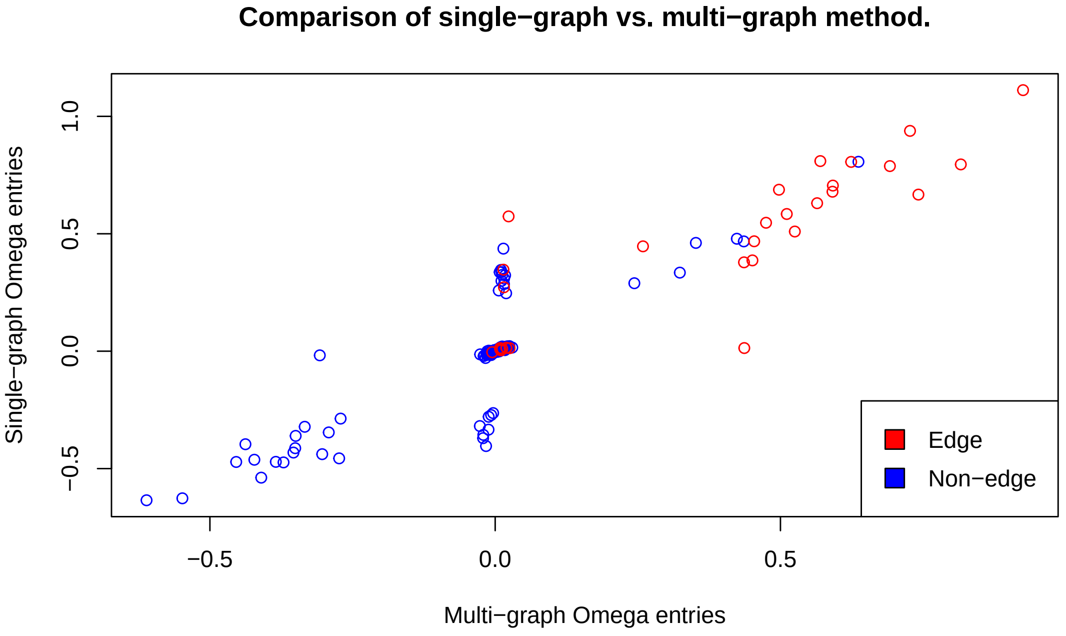

Figure 5.2: Scatterplot of single-graph estimate against multi-graph estimate, for $n = 50$, $p = 20$, and $K = 5$.

Table 5.3: F1 scores for $n = 50$ and $p = 20$. The leftmost two columns (M+B, and SG) are single-graph methods, only estimating the first graph out of K. Column “M+B” denotes Meinshausen and Buehlmann (2006) method. Column “SG” denotes the single-graph method from Z. R. Li and McCormick (2017). Columns “MG” denote the multigraph method from Chapter 3. The rows correspond to how the graph was generated. Random means that each edge indipendently had a given probability of existing. Scale-free means that the graph was created by adding one node with one edge at a time, and the edge was connected to one existing node with probability proportional to its degree. Cluster means that the nodes were separated into groups and nodes in the same group had greater chance of being connected. Each entry shows the average over 50 runs, with its standard error.

Graph	M+B	SG	MG ($K = 1$)
Random	0.56 ± 0.01	0.43 ± 0.01	0.57 ± 0.01
Scale-free	0.53 ± 0.01	0.38 ± 0.01	0.55 ± 0.01
Cluster	0.57 ± 0.005	0.5 ± 0.01	0.60 ± 0.01

5.4 Oddities regarding the likelihood, to be explored for the final version

When doing our experiments we noticed that the $Q(\Omega, \theta, \Sigma_0)$ function from Chapter 3 is not getting optimized. This is strange given that the algorithm does produce satisfactory results. We have found that the objective function will increase very quickly in the beginning iterations, but then it will decrease and settle at a lower value. This can be seen in Figure 5.3. In particular the decrease is sharper and reaches a lower point the larger the number of graphs we are using. This happens even when we do not use a prior on Σ_0. In the best case scenario it is simply a miscalculation in our code. Otherwise, this could possibly be due once again to how we generate our data, or worse, it could mean that the model is misspecified and some other approach would be more beneficial (although unlikely). Unfortunately, we do not have a particularly convincing explanation of why this phenomenon happens.

Table 5.4: F1 scores as the dissimilarity between graphs increases, for $n = 50$, $p = 20$, and $K = 4$ (refer to Subsection 5.2.1). In particular, if $prob = 0.0$, we obtain the same graph several times. The rows correspond to how the graph was generated. Random means that each edge indipendently had a given probability of existing. Scale-free means that the graph was created by adding one node with one edge at a time, and the edge was connected to one existing node with probability proportional to its degree. Each entry averaged over five runs, in which the same five arbitrarily chosen seeds were used.

Graph	$prob = 0.05$	$prob = 0.1$	$prob = 0.2$	$prob = 0.5$	$prob = 1.0$
Random	0.634	0.646	0.638	0.656	0.642

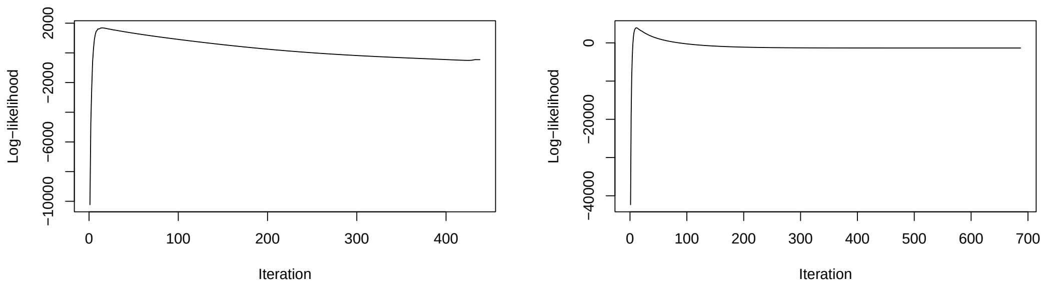

Figure 5.3: Evolution of log likelihood over iterations of fitting the multi-graph model, $p = 20$, and $n = 50$. On top we have $K = 4$ graphs, on the bottom $K = 15$.

Chapter 6

Discussion and further work

We have derived a conditional expectation maximization algorithm for inference on multiple graphs which outperforms current implementations. Using Bayesian priors, we were able to pool information across graphs to improve our estimates. For future work, we would like to solve the strange inconsistencies found during the simulations, or at least to have an explanation of why they happen. Also we would like to apply our estimation procedure to a real dataset. The code for the simulations can be found on `https://github.com/jkasalt/pdm_summary`.

For the final version of the report we will fix the code for Section 5.3, and add the third experiment. Furthermore if time permits we will apply our model to a real dataset.

Bibliography

Li, Yunfan, Bruce A. Craig, and Anindya Bhadra (July 2019). “The Graphical Horseshoe Estimator for Inverse Covariance Matrices”. en. In: *Journal of Computational and Graphical Statistics* 28 (3), pp. 747–757. DOI: 10.1080/10618600.2019.1575744. URL: http://dx.doi.org/10.1080/10618600.2019.1575744.

Li, Zehang Richard and Tyler H. McCormick (Sept. 2017). “An Expectation Conditional Maximization approach for Gaussian graphical models”. In: DOI: 10.48550/arXiv.1709.06970. eprint: 1709.06970v3. URL: https://arxiv.org/abs/1709.06970v3.

Li, Zehang Richard, Tyler H. McCormick, and Samuel J. Clark (May 2018). “Bayesian Joint Spike-and-Slab Graphical Lasso”. In: eprint: 1805.07051v2. URL: http://arxiv.org/abs/1805.07051v2.

Lukemire, Joshua, Suprateek Kundu, Giuseppe Pagnoni, and Ying Guo (Aug. 2017). “Bayesian Joint Modeling of Multiple Brain Functional Networks”. In: eprint: 1708.02123v2. URL: http://arxiv.org/abs/1708.02123v2.

Meinshausen, Nicolai and Peter Buehlmann (Aug. 2006). “High-dimensional graphs and variable selection with the Lasso”. In: Annals of Statistics 2006, Vol. 34, No. 3, 1436-1462. DOI: 10.1214/009053606000000281. eprint: math/0608017v1. URL: http://arxiv.org/abs/math/0608017v1.

Roverato, Alberto (Sept. 2002). “Hyper Inverse Wishart Distribution for Non-decomposable Graphs and its Application to Bayesian Inference for Gaussian Graphical Models”. en. In: *Scandinavian Journal of Statistics* 29 (3), pp. 391–411. DOI: 10.1111/1467-9469.00297. URL: http://dx.doi.org/10.1111/1467-9469.00297.

Wang, Hao (May 2015). “Scaling It Up: Stochastic Search Structure Learning in Graphical Models”. In: Bayesian Analysis 2015, Vol. 10, No. 2, 351-377. DOI: 10.1214/14-BA916. eprint: 1505.01687v1. URL: https://arxiv.org/abs/1505.01687v1.

Zhao, Tuo, Han Liu, Kathryn Roeder, John Lafferty, and Larry Wasserman (June 2020). “The huge Package for High-dimensional Undirected Graph Estimation in R”. In: eprint: 2006.14781v1. URL: http://arxiv.org/abs/2006.14781v1.