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

Luca Bracone

November 12, 2022

1 Introduction

1.1 Motivation

In recent times, there has been an increased interest in finding complex relationships underlying certain biological processes, such as gene expression pathways or connections between neurons in the brain. A lot of approaches in the past have focused on *directed graphical models*, in which nodes are random variables and the structure of edges force 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 on the variables a certain conditional independence structure. This report develops methods to infer edges in an undirected graph.

The Bayesian method allows specifying a prior distribution over the graphs, which can encode specific domain knowledge, depending on the field of application. For instance, the estimated graph is often chosen to be sparse, i.e. to only have few edges.

Most of 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 MonteCarlo Markov chain (MCMC) with Gibbs sampling, Wang 2015. However for most practical applications point estimates are sufficient. For this reason we will follow in the steps of Li and McCormick 2017 who derive an expectation conditional maximisation (ECM) approach to do inference. We will then build upon the method of Li and McCormick 2017, to do inference on multiple graphs rather than a single graph, using ECM. At the time of writing this report we have found a paper by Lukemire et al. 2017 whose focus is fairly similar to what we aim to do. The differences are that we will use a probit link to pool information across the graphs while they use a logistic link, and in addition we will infer the pairwise similarity between the graphs.

1.2 Basics of Bayesian statistics

Let $y_1, \ldots, y_n$ be realisations of a random distribution $p(y)$. Suppose the existence of a set of random distribution functions $P(y)$ in which $p(y)$ exists, and of a set Θ and a function $\Theta \to P(y)$ which we call a *parametrisation*. Then, statistics is concerned with using the observed $y_1, \ldots, y_n$ to estimate $\theta \in \Theta$ such that $\theta \mapsto p(y)$ is as "close" as possible to the "true" $p(y)$.

Unlike in the usual statistical context, in Bayesian statistics, we view the parameter θ as being itself random with some distribution $p(\theta)$, called *prior distribution*. We define the joint probability of y and θ as being a function $p(y, \theta)$ such that

$$p(\theta) = \int p(y, \theta) dy$$

and

$$p(y) = \int p(y, \theta) d\theta.$$

Then we define the conditional distribution "of y given θ" as

$$p(y|\theta) = \frac{p(y, \theta)}{p(\theta)}.$$

Using this property twice, we obtain Bayes' theorem which allows us to "invert" and obtain the distribution of θ given the observed y.

$$p(\theta|y) = \frac{p(y, \theta)}{p(y)} = \frac{p(\theta)p(y|\theta)}{p(y)}.$$

1.3 Bayesian hierarchical models

A hierarchical model involves prior distributions over all model parameters $\theta_j, j = 1, \ldots, p$. 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. For example, suppose that we observe values $y_{ij} \sim P(\theta_j)$ 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 are i.i.d with parameter θ_j. We will discuss here 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 estiamtor will tend to produce an estimate that has a high bias in regards to it, 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.

Finally, 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 i.i.d. distributed $\theta_j|\phi \sim Q(\phi)$, for some parameter ϕ. Then the posterior joint distribution can be expressed as

$$p(\phi, \theta|y) \propto p(y|\theta)p(\theta|\phi)p(\phi).$$

Now we need to choose a distribution $p(\phi)$ for ϕ, this is often referred to as a *hyperprior* distribution.

2 Undirected graphical models for multivariate gaussian variables

An undirected graphical model represents the conditional independence structure of some variables of interest $y_1, \ldots, y_p$ 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 $y_{-i,-j}$, for $i, j \in \{1, \ldots, p\}$. So in summary

$$y_i \not\perp\!\!\!\perp y_j | y_{-i,-j} \iff (y_i, y_j) \in E.$$

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 $\omega_{i,j}$ is zero if and only if y_i and y_j are independent given $y_{-i,-j}$. i.e.

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

2.1 Spike and slab prior for graphical models

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

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

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 $a, b, \lambda, v_0, v_1 \in \mathbb{R}$ are hyperparameters. The entries $\omega_{i,j}$ are so that the conditional distribution of Ω as a whole can be written as

$$p(\Omega|\delta) = C^{-1} \prod_{j<k} \mathrm{N}(\omega_{jk}|0, v^2_{\delta_{jk}}) \prod_j \textsc{Exp}\left(\omega_{jj}|\frac{\lambda}{2}\right) \mathbb{1}\{\Omega \in M^+\}.$$

with C a constant that depends on $\delta, v_0, v_1, \lambda$. 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 $\omega_{i,j}$ is truly zero, it is absorbed in the spike and will be estimated as close to zero. The discrete spike-and-slab instead uses a point mass at zero, $\mathbb{1}\{\omega_{i,j} = 0\}$. 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 i.i.d. observations of y.

Given this, we seek values of Ω, δ, π that maximise the log posterior joint distribution $\log\{p(\Omega, \delta, \pi|Y)\}$. The posterior joint distribution can be decomposed as

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

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

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

2.2 Expectation Maximisation for Bayesian graphical models

Following the approach in Li and McCormick 2017 instead of maximising the expression in (2) we iteratively maximise its expectation over δ. Taking the expectation of (2) we obtain

$$Q(\Omega, \pi|\Omega^{(l)}, \pi^{(l)}, Y) = \mathbb{E}_{\delta|\Omega^{(l)},\pi^{(l)},Y}\left[\log\{p(\Omega, \delta, \pi|X)\}\,\middle|\,\Omega^{(l)}, \pi^{(l)}, Y\right]. \tag{3}$$

Where $\Omega^{(l)}$ and $\pi^{(l)}$ denote the values obtained for Ω and π at the l-th iteration of the algorithm, respectively. Equation (3) is equal to

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

where $\mathbb{E}_{\delta_{ij}|\cdot}$ denotes the conditional expectation with respect to $\delta|\Omega^{(l)}, \pi^{(l)}, Y$. The expectation terms can be computed in the following way

$$q_{ij} := \mathbb{E}_{\delta_{ij}|\cdot}(\delta_{ij}) = p(\delta_{ij} = 1|\omega_{ij}^{(l)}, \pi^{(l)}) = \frac{\pi^{(l)} p(\omega_{ij}^{(l)}|\delta_{ij} = 1)}{\pi^{(l)} p(\omega_{ij}^{(l)}|\delta_{ij} = 1) + (1 - \pi^{(l)}) p(\omega_{ij}^{(l)}|\delta_{ij} = 0)} \tag{4}$$

and

$$d_{ij} := \mathbb{E}_{\delta_{ij}|\cdot}\left(\frac{1}{v_0^2(1-\delta_{ij}) + v_1^2\delta_{ij}}\right) = \sum_{\delta_{ij}=0}^{1} \frac{p(\delta_{ij}|\omega_{ij}^{(l)}, \pi^{(l)})}{v_0^2(1-\delta_{ij}) + v_1^2\delta_{ij}} = \frac{q_{ij}}{v_1^2} + \frac{1-q_{ij}}{v_0^2}. \tag{5}$$

This is the *expectation step* (E step). Now we use (4) and (5) to compute the next iterates $\pi^{(l+1)}$ and $\Omega^{(l+1)}$. The derivative of (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 Wang 2015 the authors show that that if we slice the matrices Ω, Y^tY and $V = (v_{\delta_{ij}})_{ij}$ 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 following conditional distributions

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

and

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

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

$$\omega_{12}^{(l+1)} = -((s_{22} + \lambda)\Omega_{11}^{-1} + \mathrm{diag}(d_{12}))^{-1} s_{12}$$
$$\omega_{22}^{(l+1)} = \frac{n}{s_{22} + \lambda} + (\omega_{12}^{(l+1)})^t \Omega_{11}^{-1} \omega_{12}^{(l+1)}$$

3 Extending to multiple graphs

We now have the following hierarchical model

$$\begin{aligned}
y_k|\Omega_k &\sim \mathrm{N}_p(0, \Omega_k^{-1}), \\
\omega_{ijk}|\delta_{ijk} &\sim \delta_{ijk}\mathrm{N}(0, v_1^2) + (1-\delta_{ijk})\mathrm{N}(0, v_0^2) \quad i \neq j, \quad v_0^2 \ll v_1^2 \\
\omega_{iik} &\sim \textsc{Exp}(\lambda_k/2), \\
\delta_{ijk}|\theta_{ijk} &\sim \textsc{Bern}(\Phi(\theta_{ijk})), \\
\theta_{ij} &\sim \mathrm{N}_K(0, \Sigma).
\end{aligned}$$

Where $v_0, v_1, \lambda_k, \Sigma$ are hyperparameters. To do

- Re-derive ECM algorithm for this setting

4 Choosing hyperparameters

todo!

5 Numerical techniques to make ECM more stable

todo!

6 Simulations

We perform a set of simulations with data generated from the R package `huge` (Zhao et al. 2020), we compare the results of our method with the one achieved by Meinshausen and Bühlmann 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 $l1$ penalty term. We only use one replicate but plan to add more so we can obtain measure of uncertainty for the performance of our method. We perform the tests on different graph structures that `huge` allows us to generate. In Table 1 and 2, the lines labelled by "random" indicate that the underlying graph was generated such that each edge had an equal probability. 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. To obtain the belief that a given edge δ_{ij} exists, we use (4) with the estimated Ω and π. The method by Meinshausen and Bühlmann 2006 estimates Ω as a function of η. Higher η values increase the number of zero entries in their estimate of Ω. We compare maximal *F1 scores* which is the harmonic mean between *precision* and *recall*. Where the precision

Table 1: $n = 50$, $p \in \{25, 50, 100\}$, "cheating".

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 2: $n = 200$, $p \in \{25, 35, 50\}$, more honest.

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

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. These terms are also known as positive predictive value and true positive rate, respectively. We fixed $a = b = \lambda = 1, v_1 = 100$ and tried varying values of v_0 between $1e-4$ and $1e-3$. A better approach will need to be developed in the future. The choice of v_0 and v_1 has an impact on the performance of the algorithm. Indeed, Table 1 is labelled with "cheating" meaning that we selected the "best" v_0 using the ground truth. Table 2 provides a more "honest" approach in which we picked v_0 by maximising the posterior distribution. However this often produced degenerate selections with the posterior probabilities of inclusion $p(\delta_{ij}|\Omega, \pi, Y)$ collapsing to either zero or one.

Once we have derived the ECM algorithm for multiple graphs we will also do a variety of simulations in that setting. Also we will display ROC curves, and other interesting plots once we find a good "honest" way to select the hyperparameters. The code for the simulations can be found on `https://github.com/jkasalt/pdm_summary`.

7 Application to gene-related dataset

todo!

8 Conclusion

todo!

References

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.

Lukemire, Joshua et al. (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 Bühlmann (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.

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 et al. (June 2020). "The huge Package for High-dimensional Undirected Graph Estimation in R". In: eprint: 2006.14781v1. URL: http://arxiv.org/abs/2006.14781v1.