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

Luca Bracone

November 28, 2022

0.1 Introduction

0.1.1 Motivation

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.

The Bayesian method requires the specification of 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.

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, (Wang 2015). However, for most practical applications point estimates are sufficient, so we will follow 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 "forns"(?) 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 infer the pairwise similarity between the graphs.

0.1.2 Basics of Bayesian statistics

Let $y_1, \ldots, y_n$ random variables whose distribution is $p(y)$. Suppose the existence of a set of random distribution functions $P(y)$ containing $p(y)$, and of a set Θ and a function $\varphi : \Theta \to P(y)$ that we call a *parametrisation*. Then, statistics is concerned with using the observed $y_1, \ldots, y_n$ to estimate $\theta \in \Theta$ such that $\varphi(\theta)$ 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 distribution 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

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

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

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

0.1.3 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

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

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

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

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

0.2.1 Spike and slab prior for graphical models

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

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

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

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

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 identically and independently distributed observations of y.

Given this, 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

$$p(\Omega, \delta, \pi \mid Y) = p(\Omega \mid \delta) p(\delta \mid \pi) p(Y \mid \Omega) p(\pi) p(Y)^{-1}. \tag{0.1}$$

The factor $p(Y)^{-1}$ is constant, and hence has no influence on the maximisation. Using the definitions and the decomposition of (0.1) we find that $\log p(\Omega, \delta, \pi \mid Y)$ equals

$$\begin{aligned}
\text{constant} + \sum_{i<j} \left[-\log\left\{v_0^2(1-\delta_{ij}) + v_1^2\delta_{ij}\right\} - \frac{\omega_{ij}^2}{2} \frac{1}{v_0^2(1-\delta_{ij}) + v_1^2\delta_{ij}} \right] - \sum_i \frac{\lambda}{2}\omega_{ii} \\
+ \sum_{i<j} \left\{\delta_{ij}\log(\pi) + (1-\delta_{ij})\log(1-\pi)\right\} \\
+ (a-1)\log(\pi) + (b-1)\log(1-\pi) + \frac{n}{2}\log\det(\Omega) - \frac{1}{2}\operatorname{tr}(Y^t Y \Omega).
\end{aligned} \tag{0.2}$$

0.2.2 Expectation Maximisation for Bayesian graphical models

Following Li and McCormick 2017 instead of maximising (0.2) we iteratively maximise its expectation over δ. Taking the expectation of (0.2) we obtain

$$Q(\Omega, \pi \mid \Omega^{(l)}, \pi^{(l)}, Y) = \mathbb{E}_{\delta|\Omega^{(l)},\pi^{(l)},Y}\left\{\log p(\Omega, \delta, \pi|X) \;\middle|\; \Omega^{(l)}, \pi^{(l)}, Y\right\}. \tag{0.3}$$

Where $\Omega^{(l)}$ and $\pi^{(l)}$ denote the values obtained for Ω and π at the l-th iteration of the algorithm, respectively. Equation (0.3) is equal to

$$\begin{aligned}\text{constant} - \sum_{i<j} \frac{\omega_{ij}^2}{2} \mathbb{E}_{\delta_{ij}|\cdot}\left(\frac{1}{v_0^2(1-\delta_{ij}) + v_1^2\delta_{ij}}\right) - \sum_i \frac{\lambda}{2}\omega_{ii} \\ + \frac{p(p-1)}{2}\log(1-\pi) + \sum_{i<j}\mathbb{E}_{\delta_{ij}|\cdot}(\delta_{ij})\log\left(\frac{\pi}{1-\pi}\right) \\ + (a-1)\log(\pi) + (b-1)\log(1-\pi) + \frac{n}{2}\log\det(\Omega) - \frac{1}{2}\operatorname{tr}(Y^tY\Omega),\end{aligned} \tag{0.4}$$

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

$$\mathbb{E}_{\delta_{ij}|\cdot}(\delta_{ij}) = p\left(\delta_{ij} = 1 \mid \omega_{ij}^{(l)}, \pi^{(l)}\right) = \frac{\pi^{(l)} p\left(\omega_{ij}^{(l)} \mid \delta_{ij} = 1\right)}{\pi^{(l)} p\left(\omega_{ij}^{(l)} \mid \delta_{ij} = 1\right) + (1-\pi^{(l)}) p\left(\omega_{ij}^{(l)} \mid \delta_{ij} = 0\right)} \tag{0.5}$$

which we denote q_{ij}, and

$$d_{ij} := \mathbb{E}_{\delta_{ij}|\cdot}\left(\frac{1}{v_0^2(1-\delta_{ij}) + v_1^2\delta_{ij}}\right) = \sum_{\delta_{ij}=0}^{1} \frac{p\left(\delta_{ij} \mid \omega_{ij}^{(l)}, \pi^{(l)}\right)}{v_0^2(1-\delta_{ij}) + v_1^2\delta_{ij}} = \frac{q_{ij}}{v_1^2} + \frac{1-q_{ij}}{v_0^2}. \tag{0.6}$$

This is the *expectation step* (E step). Now we use (0.5) and (0.6) to compute the next iterates $\pi^{(l+1)}$ and $\Omega^{(l+1)}$. The derivative of (0.3) 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 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 conditional distributions

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

and

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

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

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

0.3 Extending to multiple graphs

We now have the following hierarchical model

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

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

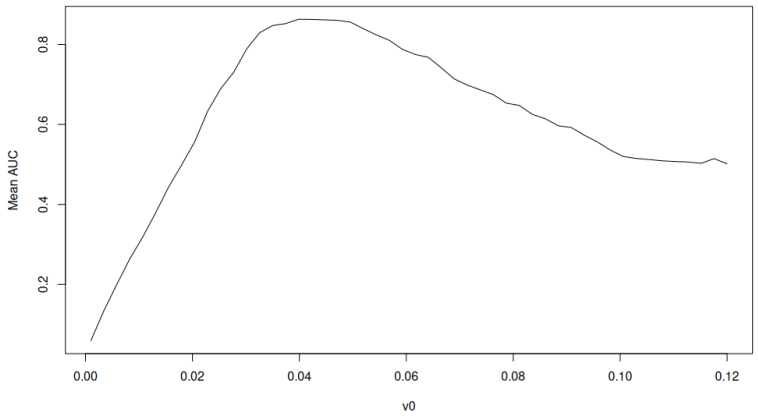
- Re-derive ECM algorithm for this setting

0.4 Choosing hyperparameters

The performance of the model is heavily influenced by the values of the hyperparameters. On Figure 0.4 we have plotted how the area under curve (AUC) changes for different values of v_0. In the single-graph case we decide to to fix $v_1 = 100, \lambda = 2$, and $a = 1$. Then we count the number of off-diagonal entries in the empirical precision matrix $\bar{\Omega}$ that are outside the interval $[-t, t]$, for $t \in \mathbb{R}$. This gives a rough estimate of the expected number of edges, s. We choose b so that the mean of the Beta distribution, $a/(a+b)$, is equal to the proportion of expected edges $2s/\{p(p-1)\}$. We then run the ECM algorithm with a different choice of v_0 each time. We pick the value of v_0 for which the output Ω has proportion of edges closest to our estimated proportion.

As we can see, the choice of t is critical as it will decide the sparsity level. To choose t there are two approaches. First, we can ourselves choose the expected sparsity level and pick t accordingly. Otherwise we can look at a histogram of the off-diagonal values of $\bar{\Omega}$ and choose a value of t where the spike and the slab seem to "cross" each other.

Figure 1: Mean AUC over ten replicates for different values of v_0, with a graph with 25 nodes and 100 samples.



0.5 Numerical techniques to make ECM more stable

todo!

0.6 Simulations

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 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 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 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. We use (0.5) with the estimates to obtain the posterior probabilities of inclusion of all edges (y_i, y_j). 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 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 the

Table 1: $n = 50$, $p \in \{25, 50, 100\}$, F1 scores. For EMGS, hyperparameters were selected by maximising F1 score.

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

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

positive predictive value and true positive rate. 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, in Table 1 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 $p(\delta_{ij} \mid \Omega, \pi, Y)$ 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`.

0.7 Application to gene-related dataset

todo!

0.8 Conclusion

todo!

Bibliography

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, 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 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, 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.