Learning the Structure of Multiple Undirected Bayesian Graphical Models with EM

Luca Bracone

November 10, 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 levels or the structure of connections between neurons in the brain. A lot of approaches in the past have focused on what are known as *directed graphical models* in which nodes are random variables and the existence of edges force the joint distribution to factor in a certain way. Some approaches have instead focused on *undirected graphical models* in which the modes are also some variables of interest, but in which the existence of edges imposes on the variables a certain conditional independence structure that we will make more precise in Section 3. Most of the work that we will do in the rest of the report will involve developing methods to infer whether or not a given edge exists in the graph.

An advantage of using the Bayesian method is the ability to specify a prior distribution over the graphs so that it can encode specific domain knowledge, depending on the field of application. For instance, it is often the case that we want the estimated graph to be sparse in the number of edges.

Most of the approaches regarding inference of undirected bayesian graphs have focused on stochastic methods to obtain an estimate of the whole posterior distribution using numerical sampling methods such as Montecarlo Markov chain (MCMC) with Gibbs sampling, see [Hao15]. However for most practical applications it is often sufficient to give a single point estimate rather than the whole distribution. For this reason we will follow in the steps of [ZT17] who derive an expectation conditional maximisation (ECM) approach to do inference. The ultimate goal for us would be to build upon the method of [ZT17], to find a way 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 [Luk+17] whose focus is fairly similar to what we aim to do. The main difference that we see is that we will use a probit link to pool information across the graphs while they use a logistic link.

1.2 Basics of Bayesian statistics

We are given a set of points $y_1, \ldots, y_n$ and imagine they are realisations of a random distribution $p(y)$. We would like to know how $p(y)$ looks like. A commonly used method is the following: suppose that there is a set of possible random distribution functions $P(y)$ in which $p(y)$ exists. Then we assume that there exists a set Θ and a function $\Theta \to P(y)$ which we call a *parametrisation*. Usually, Θ is a simpler set to study than $P(y)$. Then, statistics is concerned with using the observed $y_1, \ldots, y_n$ to draw a $\theta \in \Theta$ such that $\theta \mapsto p(y)$ is as "close" as possible to the "true" $p(y)$.

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

$$p(\theta) = \int p(y, \theta) dy$$
$$\text{and } p(y) = \int p(y, \theta) d\theta$$

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. Since we only observe a finite number of y this is only an approximate distribution, we will talk about uncertainty later on.

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

In practice, the factor $p(y)$ is nothing to worry about since we can obtain it by integrating away θ in the following way, $p(y) = \int p(y|\theta)p(\theta)d\theta$.

2 Bayesian hierarchical models

A hierarchical model is one in which instead of giving a prior $p(\theta)$ for θ we will instead model θ as being conditionally distributed on some other variable, say ϕ. Then we will try to obtain posterior estimates for both θ and ϕ. Of course, this process of "adding layers" can be repeated an arbitrary number of times but the higher a parameter is placed up the hierarchy, the higher the number of samples we need to have to produce confident estimates. We will illustrate what we have explained above with a simple example. Suppose we observe values $y_{ij} \sim P(\theta_j)$ where $j = 1, \ldots, J$ and $i = 1, \ldots, n_j$. So, in other words 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 eachother and now we can perform usual maximum likelihood estimation. Using this method may cause a problem if there is a group with only a few outlying observations. If this happens, the maximum likelihood estiamtor will tend to ignore our outlying group because it only has a few observations. Then, our outlying group will have an estimate that is far from its observations.

Then another idea might be to treat each group as a separate estimation, as though they have nothing to do with eachother. In some cases, this is justified, but it poses another problem for our outlying group with only a few observations. In this case, since that group only has a few observations we will get estimates with a large variance.

Finally, the idea of hierarhcical models is to see the first two methods as two extremes: "the θ_j are the same" and "the θ_j are the same". Would there be a way to automatically decide how similar or how different the θ_j should be from each other? Yes, instead we will imagine that the θ_j are themselves distributed $\theta_j \overset{\text{i.i.d.}}{\sim} Q(\phi)$. For some fixed 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 the *hyperprior* distribution. Once this choice is made we look for a value of θ which maximises

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

We notice that we still need to evaluate $p(\phi|y)$ often it will be possible simply to integrate away θ in the posterior joint distribution

$$p(\phi|y) = \int p(\theta, \phi|y)d\theta$$

3 Undirected graphical models for multivariate gaussian variables

An undirected graphical model is one in which the conditional independence structure of some parameters of interest $y_1, \ldots, y_p$ is represented by 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

$$y_i \perp\!\!\!\perp y_j | y_{-i,-j} \iff (y_i, y_j) \in E.$$

In the context in which we observe a sample of n p-dimensional multivariate gaussian variables $y^1, \ldots, y^n \sim N(0, \Sigma)$ we would like to deduce the structure of the graph G. The main idea here is that in the *precision matrix* $\Omega = \Sigma^{-1}$, a certain entry $\omega_{i,j}$ is zero if and only if y_i and y_j are independent given $y_{-i,-j}$. So with regard the edges of our graph G,

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

This motivates the study of the precision matrix Ω

3.1 Spike and slab prior for graphical models

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

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

where $\mathrm{N}_p(0, \Omega^{-1})$ is the multivariate normal distribution with mean 0 and covariance matrix Ω^{-1}. Beta is the usual beta distribution. The entries $\omega_{i,j}$ are independent from each other so that the conditional distribution of Ω as a whole can be written as

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

With C being some proportionality constant that depends on $\delta, v_0, v_1, \lambda$. This is known as the *spike and slab* prior because it corresponds to an overlapping of two gaussian curves, one with a small variance v_0^2 the spike and one with a large variance v_1^2 the slab. This is to represent the idea that if an entry $\omega_{i,j}$ is truly zero, when we are going to observe it, it is very likely that it will not be zero exactly but rather some random value close to zero. Another definition of spike and slab rather than using $N(0, v_0^2)$, instead uses a single point $\alpha_0 I(\omega_{i,j} = 0)$. 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 would like to obtain values for Ω, δ, π that maximise the log posterior joint distribution $\log(p(\Omega, \delta, \pi|Y))$. After a few manipulations, the posterior joint distribution can be decomposed as

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

and in practice the factor $p(Y)^{-1}$ is left out since for us it is constant, and so it has no influence on the maximisation. Using the definitions and the decomposition above we find that $\log(p(\Omega, \delta, \pi|Y))$ equals

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

3.2 Exepectation Maximisation for graphical Bayesian models

Following the approach in [ZT17] instead of maximising the expression in eq. (1) we decide to iteratively maximise its expectation over δ. So now, taking expectations of eq. (1) we obtain

$$Q(\Omega, \pi|\Omega^{(l)}, \pi^{(l)}, Y) = \mathbb{E}_{\delta|\Omega^{(l)},\pi^{(l)},Y} \left(\log[p(\Omega, \delta, \pi|X)] \,\Big|\, \Omega^{(l)}, \pi^{(l)}, Y \right). \tag{2}$$

Which is equal to

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

The expectation terms can be computed in the following way

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

and

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

Calculating these values will be called the *expectation step.* Now we use the values we have found to compute what the next iterates $\pi^{(l+1)}$ and $\Omega^{(l+1)}$ should be. The derivative of eq. (2) 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 will be slightly more complicated. Indeed we require that Ω remains positive definite after each iteration. In [Hao15] the authors show that that if we slice the matrices Ω, X^tX and $V = (v_{\delta_{ij}})_{ij}$ in the following way

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

so that ω_{22} is a scalar and ω_{12} is a $(p-1)$-dimensional vector (likewise for s_{22} and s_{12}), then 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 that $E(v_{12}^{-1}) = d_{12}$, where d_{12} is the appropriate vector of d_{ij} values. Taking the mode of these distributions motivates the update steps

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

4 Extending to multiple graphs

We now have the following hierarchical model

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

To do

- Re-derive ECM algorithm for this setting

5 Choosing hyperparameters

todo!

6 Numerical tricks to make ECM more stable

todo!

7 Simulations

We perform a set of simulations with data generated from the R package `huge` [Zha+20], we compare the results of our method with [MB06] in the following way. We only do one round of testing, so ideally in the future it would be good to do many so that we can

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

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

obtain some sort of empirical variance of how well our method performs. We perform the tests on a variety of graphs that `huge` allows us to generate. In table 1 and table 2 the lines labelled by “random” indicate that the underlying graph was generated such that each edge had an equal probability of existing. 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. Both of the methods, EMGS and huge produce a matrix with values between 0 and 1 representing how much they believe each edge should be included or not. We compare maximal *F1 scores* which is the harmonic mean between *precision* and *recall*. Where precision denotes the fraction of truly existing edges among the ones that our test declares as existing, and recall is the fraction of existing edges which our test detects. Those terms are also known as positive predictive value and true positive rate, respectively. We fixed $a = b = \lambda = 1$, then we fixed $v_1 = 100$ and tried varying values of v_0 between $1e-4$ and $1e-3$. Doing things this way is questionable, and a better approach will have to be developed in the future. A highly nontrivial question is the choice of hyperparameters λ, a, b, v_0 and v_1. In particular regarding the choice of the last two, selecting the correct values has a noticeable impact on the performance of the algorithm. Indeed, one can see that Table table 1 is labelled with cheating. It is because in that setting we were selecting the best v_0 by looking at the ground truth, so that we could convince ourselves that it was possible that this approach would work well for some choice of hyperparameters. However the more honest approach was the one in which we picked v_0 by looking at the posterior distribution, choosing the one that would maximise it. What ended up happening is that the posterior would give good results if the choice of v_0 were either really small or really big, compared to the true best one we would find by cheating. Those cases both correspond to either the case in which we select almost every edge or the one in which we select no edge, respectively. The score 0.67 may appear good but it is the one that the algorithm obtains by doing the dumb choice of not picking any edge, so we may have to pick another way to measure how good a given method performs, maybe AUC rather than maximal F1 score.

8 Application to gene-related dataset

todo!

9 Conclusion

todo!

References

[Hao15] Wang Hao. "Scaling It Up: Stochastic Search Structure Learning in Graphical Models". In: (May 2015). Bayesian Analysis 2015, Vol. 10, No. 2, 351-377. DOI: 10.1214/14-BA916. eprint: 1505.01687v1. URL: https://arxiv.org/abs/1505.01687v1.

[Luk+17] Joshua Lukemire et al. "Bayesian Joint Modeling of Multiple Brain Functional Networks". In: (Aug. 2017). eprint: 1708.02123v2. URL: http://arxiv.org/abs/1708.02123v2.

[MB06] Nicolai Meinshausen and Peter Bühlmann. "High-dimensional graphs and variable selection with the Lasso". In: (Aug. 2006). Annals of Statistics 2006, Vol. 34, No. 3, 1436-1462. DOI: 10.1214/009053606000000281. eprint: math/0608017v1. URL: http://arxiv.org/abs/math/0608017v1.

[Zha+20] Tuo Zhao et al. "The huge Package for High-dimensional Undirected Graph Estimation in R". In: (June 2020). eprint: 2006.14781v1. URL: http://arxiv.org/abs/2006.14781v1.

[ZT17] Li Zehang Richard and McCormick Tyler H. "An Expectation Conditional Maximization approach for Gaussian graphical models". In: (Sept. 2017). DOI: 10.48550/arXiv.1709.06970. eprint: 1709.06970v3. URL: https://arxiv.org/abs/1709.06970v3.