Advanced Probabilistic Machine Learning and Applications

Caterina De Bacco and Diego Baptista

January 10, 2022

Contents

1 Tutorial 8: The Stochastic Block Model and the degree-corrected SBM

1.1 Exercise 1: implementing various inferences for the standard SBM

In this tutorial we will implement various inference techniques and models to solve the SBM on real networks. We will use several codes developed in the package *pysbm* that can be found at https://github.com/funket/pysbm. This python module contains several objective functions and inference procedures, including some of those seen in Lecture 8.

(a) Clone the github repository **pysbm**.

(b) Download the datasets of **American College football** *(football)*, **Zachary's karate club** *(karate)* and **Political blogs** *(polblogs)* from http://www-personal.umich.edu/~mejn/netdata/ and put them inside the folder *pysbm/Network Data/*.

(c) Run three different inference procedures using the weighted SBM, i.e. the model with the Poisson likelihood. We suggest to run the greedy algorithm proposed by Karrer and Newman (2011) and two versions of a Monte Carlo Metropolis-Hasting scheme proposed by Peixoto (2014). Comment on their differences.

(d) Plot the adjacency matrices ordered by the inferred blocks and compare with the unordered one.

(e) Plot the affinity matrices of two partitions at your choice.

Solution in the jupyter file *L8_tutorial_solution.ipynb*.

1.2 Exercise 2: degree-corrected SBM (DC-SBM)

As you could notice in the previous exercise, the best partition found by the algorithms favours a block division correlated with degree.
In fact, maximizing the KL divergence between the SBM probability and a random uniform null model $p_0(r,s)$ as the one seen in the Lecture 8 encourages the optimal blocks to be correlated to the degree of a node, which is quite unrealistic. In other words, blocks are made of nodes of similar degree. The solution to this problem is to incorporate explicitly degree heterogeneity into the model as in the so called *degree-corrected* SBM introduced in Karrer and Newman (2011). This implies introducing new hidden variables $\theta_i \in \mathbb{R} \geq 0$ controlling the expected degree of node i. It works as follows:

$$\begin{aligned} P(\mathbf{A}|\theta,q,C) &= \prod_{i<j} \mathrm{Pois}\left(A_{ij};\theta_i\theta_j\,C_{q_iq_j}\right) & (1)\\ &= \prod_{i<j} \frac{e^{-\theta_i\theta_j\,C_{q_iq_j}}\;\left(\theta_i\theta_j\,C_{q_iq_j}\right)^{A_{ij}}}{A_{ij}!} \quad . & (2) \end{aligned}$$

One can normalize this new parameter as:

$$\sum_i \theta_i \delta_{q_i,r} = 1 \quad \forall r = 1,\dots,K \quad . \tag{3}$$

Then θ_i can be interpreted as the probability that an edge connected to the group q_i lands to i itself.

(a) Derive the null model suited for the KL divergence representation of the DC-SBM as done in the Lecture 8 for the standard SBM.
Comment on it.

(b) Run the same inference as before, but this time using the degree-corrected likelihood as objective function.
Comment on the different partitions obtained compared to the standard SBM.

(a) We can rewrite eq. (2) as

$$P(\mathbf{A}|\theta,q,C) = \prod_i \theta_i^{k_i} \prod_{r<s} \frac{e^{-C_{rs}}\,C_{rs}^{m_{rs}}}{A_{ij}!} \quad , \tag{4}$$

and the logarithm form is

$$\log P(\mathbf{A}|\theta,q,C) = 2\sum_i k_i \log\theta_i + \sum_{rs}(m_{rs}\log C_{rs} - C_{rs}) \quad . \tag{5}$$

Allowing for the constraint in eq. (3), the MLE for θ_i and C_{rs} are

$$\theta_i = \frac{k_i}{k_{q_i}} \quad , \quad C_{rs} = m_{rs} \quad , \tag{6}$$

where k_{q_i} is the sum of the degrees in the group where node i belongs to. For example k_r is the sum of the degrees of the vertices in group r and

$$k_r = \sum_s m_{rs} = \sum_i k_i \delta_{q_i,r} \quad . \tag{7}$$

Substituting eq. (6) into eq. (5) we obtain

$$\log P(\mathbf{A}|q,\hat{\theta},\hat{C}) = 2\sum_i k_i \log\frac{k_i}{k_{q_i}} + \sum_{rs}(m_{rs}\log m_{rs}) - 2|E| \quad . \tag{8}$$

We are still interested in giving a MLE interpretation to this expression so we rewrite the first term of the expression as

$$\begin{aligned}
2\sum_i k_i \log\frac{k_i}{k_{q_i}} &= 2\sum_i k_i \log k_i - 2\sum_i k_i \log k_{q_i} && (9)\\
&= 2\sum_i k_i \log k_i - 2\sum_i \sum_r k_i \delta_{q_i,r} \log k_r && (10)\\
&= 2\sum_i k_i \log k_i - \sum_r k_r \log k_r - \sum_s k_s \log k_s && (11)\\
&= 2\sum_i k_i \log k_i - \sum_{rs} m_{rs} \log(k_r k_s) \quad , && (12)
\end{aligned}$$

where we used the relation in eq. (7). Substituting back into eq. (8) and dropping constants we obtain the following unnomalized log-likelihood function

$$\log P(\mathbf{A}|q,\hat{\theta},\hat{C}) = \sum_{rs} m_{rs} \log \frac{m_{rs}}{k_r k_s} \quad . \tag{13}$$

Adding and multiplying by constant factors allows us to write the log-likelihood in the form

$$\log P(\mathbf{A}|q,\hat{\theta},\hat{C}) = \sum_{rs} \frac{m_{rs}}{2|E|} \log \frac{m_{rs}/2|E|}{(k_r/2|E|)(k_s/2|E|)} \quad , \tag{14}$$

where $p_{deg}(r,s) = \frac{k_r}{2|E|}\frac{k_s}{2|E|}$ is the probability distribution over the group assignments at the end of a randomly chosen edge. Thus, eq. (14) is the Kullback-Leibler divergence between $p_K(r,s)$ and $p_{deg}(r,s)$. We can conclude that the best fit to the degree-corrected stochastic blockmodel gives the group assignment that is most surprising compared to the null model with given expected degree sequence, whereas the ordinary stochastic blockmodel gives the group assignment that is most surprising compared to the Erdos-Renyi random graph.

(b) Solution in the jupyter file *L8_ tutorial_ solution.ipynb*.

1.3 Exercise 3

> Choose two inference methods and apply similar analysis for the football network (K=11) and the political blogs one (K=2).

Solution in the jupyter file *L8_ tutorial_ solution.ipynb* by changing the **dname** in cell [4].

2 More

- For an interesting review on the possible extensions of the SBM, check Lee and Wilkinson (2019).
- The weighted SBM assumes the edges are distributed as a Poisson distribution. The Poisson distribution may generate multiple edges between a pair of nodes, so this model may create the so called *multigraphs*. This is consistent with the interpretation that A_{ij} is the number, or total weight, of links from i to j. If we wish to generate binary networks where $A_{ij} \in \{0,1\}$, we use the fact that the Poisson and Bernoulli distributions become close in the sparse limit.

References

B. Karrer and M. E. J. Newman, Phys. Rev. E **83**, 016107 (2011).

T. P. Peixoto, Phys. Rev. E **89**, 012804 (2014).

C. Lee and D. J. Wilkinson, Applied Network Science **4** (2019), 10.1007/s41109-019-0232-2.