Aspects of community detection on graphs

Leonardo Martins Bianco

Advised by Christine Keribin and Zacharie Naulet

Contents

1 Introduction 2
- 1.1 Objectives of the internship . . . 2
- 1.2 Motivations for the internship . . . 2

2 Algorithms for community detection 8
- 2.1 Statistical approach . . . 8
- 2.2 Spectral approach . . . 16

3 Statistical learning theory 21
- 3.1 Agreement and degrees of recovery . . . 21
- 3.2 Asymptotic topologies . . . 22
- 3.3 Consistency of variational EM . . . 26
- 3.4 The case of of spectral clustering . . . 26

4 Numerical experiments 31
- 4.1 Numerical experiments . . . 31

5 Conclusion and outlook 32
- 5.1 Conclusions . . . 32

A Proofs, calculations, extra definitions 33
- A.1 Proof of Theorem 1 . . . 33
- A.2 Proof of Proposition 1 . . . 34
- A.3 Proof of EM explicit steps . . . 35
- A.4 Convergence of VEM to a local maximum . . . 37
- A.5 Sketch of proof of Lemma 1 . . . 38
- A.6 Asymptotic notation . . . 39

B The case of two communities 40
- B.1 The case of two communities* . . . 40

Chapter 1

Introduction

1.1 Objectives of the internship

This report succintly presents the work done during the internship concluding the Master 2 program MVA - *Mathématiques, Vision, Apprentissage*. The internship was academic in nature, and lasted six months. It will be followed by a PhD thesis on the same subject, by the same intern, under the same advisors. Its general goals were, sequentially,

1st To introduce the intern to the problem of community detection on graphs. This consists on getting a firm understanding on the different canonical approaches to the problem, gaining familiarity first with the field's classical and recent research literature, and gaining probabilistic as well as statistical intuitions that can be applied to related problems;

2nd To tackle a current research question, suggested by the advisors of the internship;

3rd To apply the knowledge obtained to think of new research directions autonomously.

1.2 Motivations for the internship

The field of community detection on graphs, also called graph clustering, is rich both in theory and in applications. In this section, the motivations for the field and for the particular problem of the internship are explained.

1.2.1 Graphs

A graph is a mathematical object expressing interactions between entities. There are many variations of graphs, such as undirected, directed, weighted, and so on.

Definition 1. A *simple graph* G is a pair of sets (V, E), where V is a set whose elements are called *vertices*, and E is a set of pairs of distinct vertices such that $(i, j) \in E$ implies $(j, i) \in E$, whose elements are called *edges*.

Remark. From now on in this report, the term *graph* will stand for *simple* graphs.

Definition 2. A *subgraph* $F = (V', E')$ of a graph $G = (V, E)$ is a graph such that $V' \subset V$ and $E' \subset E$. A parent graph $H = (\tilde{V}, \tilde{E})$ of G is a graph such that G is a subgraph of H.

It is useful to be able to talk about specific nodes of the graph using numbers.

Definition 3. Let $G = (V, E)$ be a graph on $|V|= n$ nodes. An *enumeration* of its nodes is a bijection $\iota : V \to \{1, \dots, n\}$.

Remark. In this report, enumerated nodes will still be denoted simply by V.

These enumerations allow representing the graph with a matrix. In the case of simple graphs, this matrix is symmetric, making it possible to bring a whole range of algebraic techniques to the study of graphs.

Definition 4. Let G be a graph, and assume an arbitrary enumeration of its nodes. Define the *adjacency matrix* A associated to G by

$$A_{ij} := \begin{cases} 0 & \text{if } (i, j) \notin E, \\ 1 & \text{otherwise}, \end{cases} \tag{1.1}$$

where (i, j) is a pair of nodes enumerated as i and j.

Remark. Notice that permuting the enumeration of nodes of G correspondingly permutes the associated rows and columns of the adjacency matrix.

One of the key quantities in the study of graphs is the degree of nodes.

Definition 5. Let $G = (V, E)$ be a graph. The *degree* of node i is the number of edges connected to it, that is, $d_i := |\{(k, l) \in E : k = i\}|$.

Algebraically, the degree of node i equals $d_i = \sum_{k=1}^{n} A_{ik}$, and the information about all the degrees can be gathered in a diagonal matrix.

Definition 6. Let $G = (V, E)$ be a graph, and A be its adjacency matrix. The *degree matrix* of G, denoted D, is defined as

$$D_{ij} := \begin{cases} 0 & \text{if } i \neq j, \\ \sum_{k=1}^{n} A_{ik} & \text{if } i = j. \end{cases} \tag{1.2}$$

It is also fundamental to rigorously define the notion of connectivity of graphs.

Definition 7. Let $G = (V, E)$ be a graph. A *finite walk* is a sequence of edges $(e_1, \dots, e_{n-1})$ for which there is a sequence of vertices $(v_1, \dots, v_n)$ such that for each i, $e_i = (v_i, v_{i+1})$. The finite walk is then said to *connect* v_1 to v_n.

A cycle is a finite walk in which only the first and last vertices are equal.

Definition 8. A graph is *connected* if for any pair of nodes i and j there exists a finite walk connecting i to j. Otherwise, it is said to be *disconnected*.

Definition 9. A *connected component* of G is a connected subgraph such that any parent graph to it is disconnected.

Finally, the general notion of splitting nodes of a graph into separate groups can be formalized using partitions.

Definition 10. Let S be a set. A *partition* P of S is a collection of sets such that P does not contain the empty set, the union of the sets in P equal S, and the intersection of any two distinct sets in P is empty.

Definition 11. Let $G = (V, E)$ be a simple graph. A *partition of nodes* of G is a partition of V.

1.2.2 Graphs with communities

One interesting and relevant structure a graph can have is that of *communities*.

Definition 12. Let $G = (V, E)$ be a graph. An *assignment of communities* to its nodes is a map $Z : V \to \{1, \dots, k\}$. The quantity k is the *number of communities* of this assignment.

It is clear that any assignment of communities on a graph induces a partition of its nodes: if $G = (V, E)$ is the graph, then the *community sets* $\Omega_j := \{i \in V : Z_i = j\}$, defined for $j = 1, \dots, k$, partition V.

Remark. There are models allowing nodes with multiple communities, but this escapes the scope of this report.

Even though it is easy to define what a community assignment is, precisely defining what communities *are*, in the sense of what these assignments should represent or how they should be built, is subtle and not agreed upon. One common intuition for a community is to think of it as a set of vertices having more connections among themselves than with all other vertices. Such an intuition is useful in many situations, and it is called an *assortative* notion of community. In other cases, however, one's intuition of what such groups are does not fit the assortative case. Consider a party, where people dance in pairs. There are fifty men and fifty women, and assume that each man will pair up with some woman to dance. In this case, one can consider that there are two groups in the party, men and women. However, no two members of the same group connect. This is what is called an *dissortative* notion of community, as its members share a *pattern* of connection instead of denser connections. This, coupled with many other conceptual difficulties, is the reason why the problem of community detection on graphs is delicate.

1.2.3 Common approaches to community detection

Assume an arbitrary graph is observed, with the sole hypothesis that in it there are communities, i.e., there exists some particular partition of the graph corresponding to community assignments. This hypothesis is deliberately vague. Consider the following common, yet very distinct, approaches to building algorithms to find such partition.

Statistical approaches

In *statistical* approaches to community detection, one assumes that the communities assigned to nodes and the edges formed are random variables, and thus the graph observed arises as an observation from some model. One then analyzes its likelihood function to derive algorithms derived in order to estimate the model's parameters and infer the graph's communities. In this report, the model assumed is called the stochastic block model (or SBM for short). It is a particular choice, as there are many other models, motivated by its popularity in this research field. As it will be seen, directly computing the likelihood of an observation under such model is intractable, since doing so requires one to sum over all possible cluster assignments for the nodes. The number of terms in such a sum grows exponentially with the number of nodes, and there is no way to simplify it into a tractable form. A common way around this difficulty is to substitute the exact likelihood function by a variational approximation to it. This approximation can be built in various ways, the simplest one being the *mean-field* approximation. The optimization problem that arises can in theory be solved by an alternating optimization algorithm akin to the classical EM algorithm, and is called the Variational EM.

Spectral approaches

In *spectral* approaches to community detection, it is not assumed that the graph at hand is an observation of some (probabilistic) model. Instead, one derives algorithms based on heuristics and approximations to optimization problems. One popular optimization problem for finding communities in graphs is the balanced min-cut problem, which searches for a partition such that the number of edges across classes is minimal. This agrees with the assortative intuition of what a community is. This problem is NP-hard and is popularly approximated by a relaxed version leading to the spectral clustering algorithm.

These spectral approaches can be seen as ways of embedding the graph in some vector space. In this vectorial representation of the graph, there are as many vectors as there are nodes, and as many dimensions as communities. In consequence, its dimensionality is typically low when compared to the complexity of the general graph representation. Moreover, under certain conditions, clustering the nodes in this representation (using classical vector space clustering algorithms such as k-means) could correspond to a clustering of nodes in the graph, thus revealing the communities.

1.2.4 Statistical learning theory

These approaches are popular, but without a ground truth for the communities on the graph there is no way of measuring the accuracies of their results. In such an *unsupervised* setting, the answer to the question of what are the graph's communities must be the output of the algorithm itself.

One way of dealing with these difficulties is to consider a *generative* model. This provides a definite ground truth assignment of communities to the nodes of a graph arising as an observation from such model. As a consequence, one can develop, measure the accuracy, and compare algorithms designed to recover these communities. Arguably, the most popular such model is the Stochastic Block Model (SBM). Essentially, it randomly assigns communities to nodes and then connects any pair of them with a probability depending on the communities of the pair, see Figure 2.1.

1.2.5 Applications

Data in the form of graphs and networks naturally appear in fields such as ecology, power grids, transportation systems, social networks, recommendation systems, neurology, telecommunications, and so on. Some interesting applications of community detection methods include the analysis of political blogospheres [6], analysis of criminal networks [7], cell profiling [13], analysis of ecological networks [12], and so on. There is a growing abundance of network data openly available online. Some useful resources are [14, 16, 15, 8]. This reports presents experiments performed in simulated data, as well as in real data coming from these sources.

1.2.6 Note

It is important to emphasize that this report aims to convey, beyond the effort put into answering any particular question, the view the intern ended up having of the field and the new challenges to be adressed as a continuation on his PhD. Several research problems were explored during these months. This was at times intentional, aiming to give the intern an understanding of the different open questions in the field, and at other times a result of the difficulty of the question at hand.

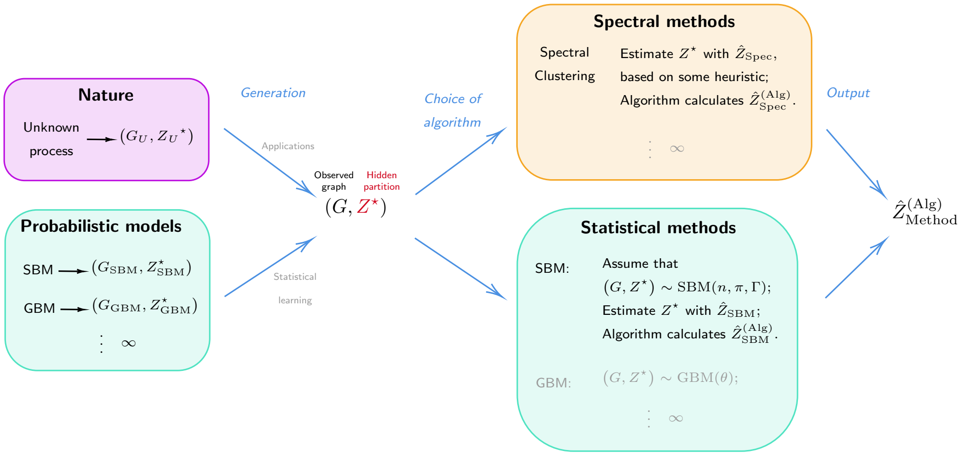

Nature
Unknown process → $(G_U, Z_U{}^\star)$
Probabilistic models
SBM → $(G_{\mathrm{SBM}}, Z^\star_{\mathrm{SBM}})$
GBM → $(G_{\mathrm{GBM}}, Z^\star_{\mathrm{GBM}})$
$\vdots\ \infty$
Generation
Applications
Statistical learning
Observed graph
Hidden partition
$(G, Z^\star)$
Choice of algorithm
Spectral methods
Spectral Clustering
Estimate $Z^\star$ with $\hat{Z}_{\mathrm{Spec}}$, based on some heuristic;
Algorithm calculates $\hat{Z}^{(\mathrm{Alg})}_{\mathrm{Spec}}$.
$\vdots\ \infty$
Statistical methods
SBM:
Assume that $(G, Z^\star) \sim \mathrm{SBM}(n, \pi, \Gamma)$;
Estimate $Z^\star$ with $\hat{Z}_{\mathrm{SBM}}$;
Algorithm calculates $\hat{Z}^{(\mathrm{Alg})}_{\mathrm{SBM}}$.
GBM:
$(G, Z^\star) \sim \mathrm{GBM}(\theta)$;
$\vdots\ \infty$
Output
$\hat{Z}^{(\mathrm{Alg})}_{\mathrm{Method}}$

Figure 1.1: Steps taken from observing a graph to estimating its communities. First, either an unknown process or some probabilistic model generates a graph, which is observed, and a community assignment on it, which is hidden (not observed). Then, an algorithm is picked to estimate the communities. This report presents two different approaches, namely spectral and statistical, to develop of such algorithms.

Chapter 2

Algorithms for community detection

This chapter presents two distinct approaches to finding communities on a graph. First, a statistical approach is developed, where a stochastic block model is assumed to have generated the graph observed and then a particular variational method is employed to estimate this model's parameters and infer the communities. Then, an alternative approach based on analyzing the spectrum of graph operators is introduced.

2.1 Statistical approach

In this section, a framework for *statistical* community detection is laid down. It consists in assuming a SBM model for the observed graph, then estimating the parameters of the model by approximately maximizing the likelihood using an algorithm similar to the classical EM, called the *variational EM algorithm*. In this process, the resulting *variational parameters*, that control the approximation of the likelihood, assign for each node in the graph a vector describing the probability of it belonging to each community; a community estimation can be obtained by taking the index of the maximal entry of this vector, for each node. Notice that this is one particular instance of statistical approach; others can be developed by choosing different models (such as latent block models, geometric block models, etc) or different estimation procedures (such as stochastic EM, variational Bayes, Gibbs sampling, and so on).

2.1.1 The stochastic block model

Part of the statistical procedure is to assume a model for the observations at hand. This subsection defines a model which is popularly assumed in the context of community detection on graphs, and which will be used in the analysis that follows.

The general SBM

One canonical probabilistic model for graphs with community structure is called the *stochastic block model*, or SBM for short. For a lengthier discussion on the origins and variants of this model, refer to [1].

Definition 13 (Stochastic blockmodel)**.** Let $n \in \mathbb{N}$, $k \in \mathbb{N}$, $\pi = (\pi_1, \dots, \pi_k)$ be a probability vector on $[k] := \{1, \dots, k\}$ and Γ be a $k \times k$ symmetric matrix with entries $\gamma_{ij} \in [0, 1]$. A pair (Z, G) is said to be *drawn under a* $\mathrm{SBM}(n, \pi, \Gamma)$ if

- $Z = (Z_1, \dots, Z_n)$ is an n-tuple of $\mathbb{N}^k$-valued random variables $Z_i \sim \mathcal{M}(1, \pi)$,
- G is a simple graph with n vertices whose symmetric adjacency matrix has zero diagonal and for $j > i$, $A_{ij}|Z \sim \mathrm{Ber}(\gamma_{Z_i,Z_j})$, the lower triangular part being completed by symmetry.

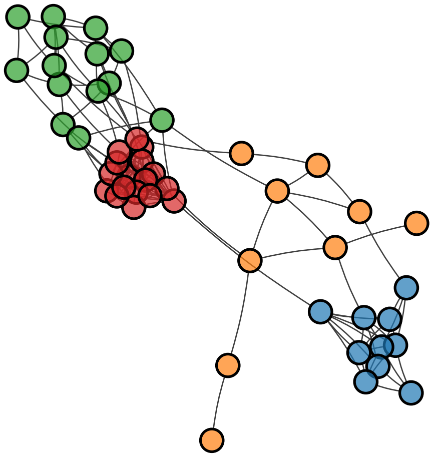


Figure 2.1: An example of an SBM graph with assortative communities

Remark. The quantity n should be thought of as being the number of nodes in G, k should be thought of as being the number of communities in G, π should be thought of as being a prior on the community assignments Z, and Γ should be thought of as a matrix of intra-cluster and inter-cluster connectivities. The random variables of the model are the n community assignments Z and the $\binom{n}{2}$ entries A_{ij} of the adjacency matrix.

Remark. Although each community assignment Z_i is a vector, the same Z_i can be used to denote the *number* of the community that node i is assigned to, to make notation lighter.

It is important to emphasize that although intuition frequently refers to the assortative case, such as in Figure 2.1, the SBM is versatile and can reproduce many other characteristics of graphs with communities. For instance, the SBM can generate bipartite graphs as a model for the example given in the introduction, where couples dance in a party; it can also generate graphs with "stars", and reproduce the "core-periphery" phenomenon. See Figure 2.2.

The symmetric SBM

Even though the SBM is a simple and intuitive model for graphs with communities, the calculations associated with it can already yield long expressions and present subtleties. For this reason, it is desirable to have a yet simpler version of the SBM where one can test intuitions and perform preliminary calculations. The symmetric SBM is precisely such a model.

Definition 14 (Symmetric SBM)**.** The pair (X, G) is drawn from SSBM (n, k, p, q) if it is drawn from an SBM model with $\pi = \frac{1}{k}\mathbf{1}_k$ and Γ taking values p on the diagonal and q outside the diagonal.

2.1.2 Model likelihood

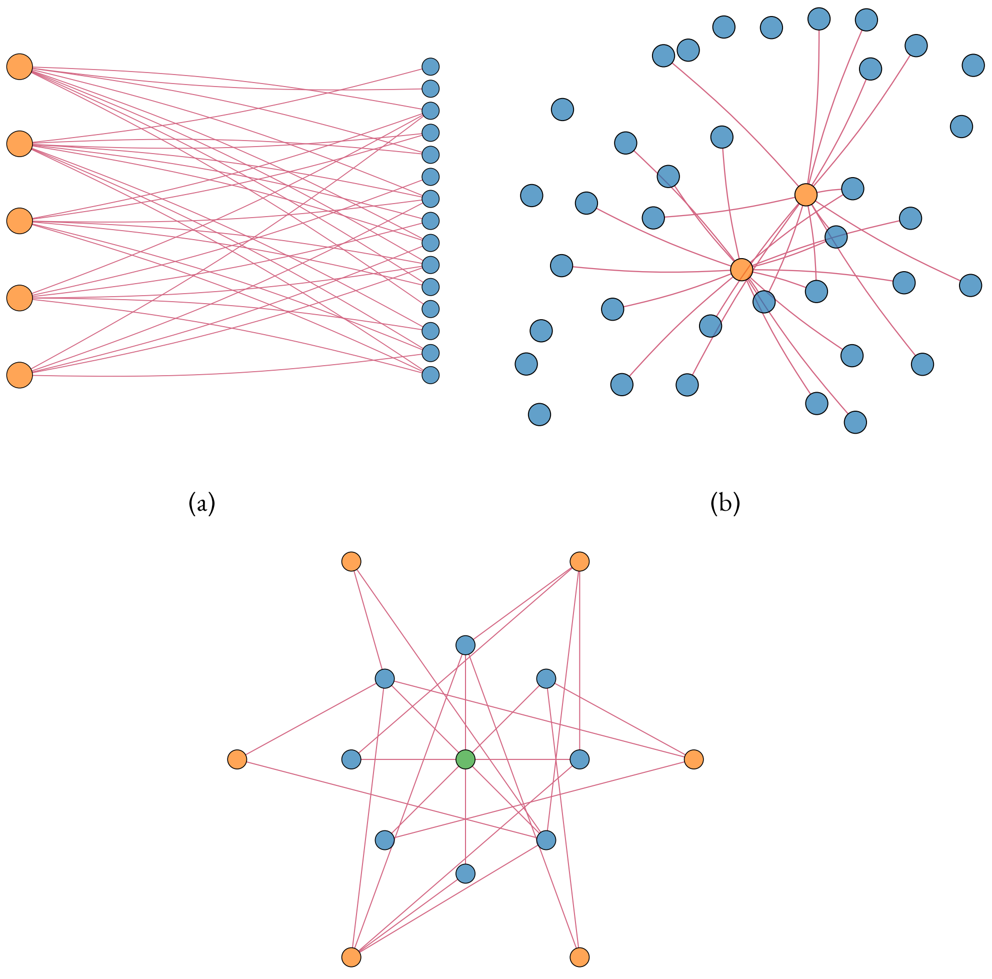
After choosing a model for one's observations, the next step in the statistical procedure is to estimate its parameters. A popular estimation procedure is to maximize *likelihood* of the observations. This subsection introduces the SBM likelihoods, while also explaining how to deal with the presence of latent variables.

Complete model likelihood

Given the observation of a graph G, represented by its adjacency matrix A, and fixing community assignments Z for it, one can calculate for each choice of model parameters $\theta := (\pi, \Gamma)$ the probability $p_\theta(A, Z)$ of observing A, Z under the model SBM(n, π, Γ). The function associating parameters $\theta \to p_\theta(A, Z)$ is called the *complete likelihood* for (A, Z). It is "complete" in the sense that it assumes knowledge of Z, which is latent (not observed in practice). It can be explicitly written as

$$p_\theta(A, Z) = \prod_{i=1}^{n} \pi_{Z_i} \prod_{\substack{i=1 \\ j>i}}^{n} \gamma_{Z_i Z_j}^{A_{ij}} (1 - \gamma_{Z_i Z_j})^{1-A_{ij}}. \tag{2.1}$$

Remark. Equation (2.1) is an example of the abuse of notation described in the remarks below Definition 13, since here Z_i denotes the *number* of the community associated to node i.


(a)

(b)

(c)

Figure 2.2: The SBM is versatile and can give rise to different features, such as (a) bipartite structures, (b) star structures, (c) core-periphery structures.

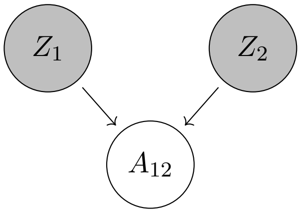


Figure 2.3: Graphical model expressing the dependence structure between latent variables Z_1, Z_2 and edge A_{12}; this pattern is commonly called a "v-structure" [5]

Observed model likelihood

In order to have a likelihood associated to an observation, one needs to marginalize the latent variables present in Equation (2.1). That is, if n is the number of nodes in the graph, and k is the number of communities, then

$$p_\theta(A) \coloneqq \sum_{Z \in k^n} p_\theta(A, Z). \tag{2.2}$$

However, in practice the calculation of this sum is intractable. First notice that there is a complex dependency structure between the Z_i given A. To see this, consider the case of $k = 2$ communities and without loss of generality enumerate any pair of nodes as nodes 1 and 2. The edge connecting them is represented in the adjacency matrix by A_{12}. The graphical model of this situation is shown in Figure 2.3. The joint probability distribution of $Z_1, Z_2 | A$ writes

$$p(Z_1, Z_2 | A_{12}) = \frac{p(Z_1, Z_2, A_{12})}{p(A_{12})} = \frac{p(A_{12} | Z_1, Z_2) p(Z_1) p(Z_2)}{p(A_{12})}. \tag{2.3}$$

which does not factorizes as $p(Z_1 | A_{12}) p(Z_2 | A_{12})$, showing the dependence of Z_1 on Z_2 (and vice-versa) conditionally on A_{12}. As a consequence, the sum cannot be simplified by writing $p(A, Z) = p(Z|A)p(Z)$. Moreover, the sum has a number of terms which is exponential on the number of nodes n, making its computation infeasible for many graphs appearing in applications.

Therefore, for most practical applications it will be necessary to approximate this observed likelihood in order to perform statistical estimation and inference on SBMs.

2.1.3 Variational decomposition and mean field approximation

The likelihood in Equation (2.2) is complex to deal with, since it has a number of terms exponential in n and also because it can have multiple local optima. Therefore, in most cases approximations are needed in order to work with this model. A common one is the variational approximation to the likelihood. This is still complicated in all its generality, so a second "mean field" approximation is used on top of the first variational one. This consists in searching the solution to the variational approximation amidst factorizable distributions. These approximations will deal with the problem of having an exponential number of complicated terms, but unfortunately the problem of multiple local optima will still remain.

Deriving the variational decomposition.

Let Z denote the vector of latent variables, A an observation, and $\theta := (\pi, \Gamma)$ the parameters of an SBM. The following *variational decomposition* leads to an useful approximation to the likelihood.

Definition 15. For any two probability distributions $p(z), q(z)$,

$$\mathrm{KL}(q\|p) := -\int_Z \log\left(\frac{p(z)}{q(z)}\right) q(z)\, dz \tag{2.4}$$

is called the *Kullback-Leibler divergence* from $p(z)$ to $q(z)$.

Theorem 1. *For any probability mass function $q(z)$ over k^n, the observed likelihood can be decomposed as*

$$\log p_\theta(A) = F(q, \theta) + \mathrm{KL}(q\|p_\theta(Z|A)), \tag{2.5}$$

where

$$F(q, \theta) := \int_Z \log\left(\frac{p_\theta\,(Z, A)}{q(Z)}\right) q(Z)\, dZ \tag{2.6}$$

is called the evidence lower bound, or ELBO for short.

For the proof, see Section A.1. Equation (2.5) forms the basis of the classical EM algorithm for estimating θ, where one performs alternate minimization of the KL term and subsequent maximization of the term $F\,(q, \theta)$. A classical result proves that the KL is always positive. Therefore, the ELBO is indeed a lower bound for the log likelihood being decomposed. Given the independence of the left hand side with respect to q, observe that maximizing the ELBO amounts to minimizing the KL term, and this can be done by setting $q = p_\theta(Z|A)$. However, in the case of the SBM, calculating expectations with respect to this conditional probability is itself intractable, due to its complex dependency structure (Equation (2.3)), so this step of EM must be performed differently.

In the mean-field case, the ELBO is also sometimes called the "free energy", due to its relevance in the physics literature.

Mean field approximation.

A common strategy used to deal with the problem of having an untractable solution q to the variational approximation is called the "mean field approximation". It consists in trying to find a q distribution maximizing the ELBO constrained to a family of tractable distributions. One possible choice for tractability is to consider factorizable distributions, that is, distributions of the form

There are other "correlated" mean field approximations, see [11].

$$q(Z) = \prod_{i=1}^{n} q_i(Z_i).$$

2.1.4 Variational estimation of the SBM

The ELBO in the SBM case

In the case of the SBM, each factor in the mean field approximation must be multinomial distribution, and they differ only by their parameters, that is,

$$q(Z) = \prod_{i=1}^{n} m(Z_i;\tau_i), \tag{2.7}$$

where $m(z,\tau_i) \propto \prod_{j=1}^{k} \tau_{ij}^{z_j}$ is the probability mass function of a multinomial distribution with parameter τ_i. Notice that $\tau_i \in [0,1]^k, \sum_j \tau_{ij} = 1$. They can be arranged as rows in a single matrix τ, and will collectivelly be called the *variational parameters*, as they control the distribution being used to approximate the posterior in the variational family. It is then possible to find an explicit form for the ELBO of an observation assuming the SBM model.

Proposition 1. *Given an adjacency matrix A and assuming an* SBM(n,π,Γ)*, the mean-field ELBO is given by*

$$F_A(\tau,\theta) = \sum_{i=1}^{n}\sum_{k=1}^{K}\left[\tau_{ik}\log\frac{\pi_k}{\tau_{ik}} + \frac{1}{2}\sum_{j=1}^{n}\sum_{l=1}^{K}\tau_{ik}\tau_{jl}\left(A_{ij}\log\gamma_{kl} + (1-\delta_{ij}-A_{ij})\log(1-\gamma_{kl})\right)\right]. \tag{2.8}$$

For the proof, see Section A.2. In practice, the dependence on A will be ommitted and the ELBO will be denoted solely by F.

The variational EM algorithm

Traditionally, optimization under missing data is numerically done via the EM algorithm. Maximizing the ELBO in models with latent variables is a particular instance of this situation. In this case, since there is the extra step of approximating the q distribution within the mean-field variational family, the algorithm is called the *variational EM*, Algorithm 1.

Algorithm 1 Variational EM

```
Require: Adjacency matrix $A$, tolerance tol
  Randomly initialize $\theta_0 = (\pi_0, \Gamma_0)$
  Randomly initialize $\tau_0$
  Initialize variation $\leftarrow 1$
  while variation $>$ tol do
      $\tau_{t+1} \leftarrow \text{argmax}_\tau \, F(q_\tau, \theta_t)$    (E step)
      $\theta_{t+1} \leftarrow \text{argmax}_\theta \, \mathbb{E}_{q_{\tau_{t+1}}}[\log p_{\theta_t}(x, z)]$    (M step)
      variation $\leftarrow F(\theta_{t+1}, \tau_{t+1}) - F(\theta_t, \tau_t)$
  end while
  Return final parameters $\theta_f$ and $\tau_f$
```

This algorithm is monotone, in the sense that each step always makes the likelihood increase; see Section A.4. Therefore, when the likelihood is bounded, the algorithm will converge to some local maximum. In practice, one performs these steps iteratively until the variation of the ELBO becomes negligible.

It is possible to explicitly describe these steps. The E step can be calculated by solving the fixed point relation

$$\hat{\tau}_{ik} \propto \pi_k \prod_{\substack{j>i \\ l=1,\dots,K}} \left(\gamma_{kl}^{A_{ij}} \left(1-\gamma_{kl}\right)^{(1-A_{ij})}\right)^{\tau_{jl}} . \tag{2.9}$$

The M step can be calculated directly by

$$\hat{\pi}_k = \frac{1}{n} \sum_{i=1}^{n} \tau_{ik}. \tag{2.10}$$

and

$$\hat{\gamma}_{kl} = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} \tau_{ik} \tau_{jl} A_{ij}}{\sum_{i=1}^{n} \sum_{j=1}^{n} \tau_{ik} \tau_{jl} \left(1 - \delta_{ij}\right)}. \tag{2.11}$$

For a proof of these statements, see Section A.3.

2.2 Spectral approach

This section brings forward a different family of methods for community detection, which in contrast to the previous statistical approach does not assume a model for the graph observed. First, the general philosophy is presented; then, the main object, graph Laplacians, are defined and their basic properties with relation to community detection are established. The main optimization problem is then written, and subsequently approximated using the graph Laplacian. An algorithm immediately ensues.

2.2.1 The spectral point of view

The use of transformations has proven to be, time and again, very useful in mathematics. Classical examples are the Fourier transform and its uses in signal processing, and the Laplace transform with its uses in differential equations and probability. The idea is always to bring a dual point of view to the problem. For example, the Fourier transform allows one to analyze a signal processing problem in the time domain or in the frequency domain, according to which one is more convenient.

The same idea applies for graphs. On the one hand, a graph can be analyzed in the *topological* domain, and quantities based on the nodes' connectivity can be calculated. Examples of topological quantities are the diameter, the degree, the clustering coefficient, the girth, but there are many others. On the other hand, the graph can be represented by a symmetric adjacency matrix A that is completely characterized by its eigensystem. This follows from the spectral decomposition for symmetric matrices, and in the case of the adjacency matrix it writes $A = X\Lambda X^t$.

E.g. "girth" is defined as the length of a shortest cycle in the graph; if there are no cycles, it is defined to be infinite.

There are reasons for which the spectral domain might present advantages for analyzing a graph. The topological quantities described are usually correlated and dependent, while the eigenvectors of A are orthogonal and its eigenvalues are independent quantities. The spectral domain representation of a graph lies in an Euclidean space, so if structures appear in it, it is possible to use classical algorithms on such spaces. These tend to be simpler, faster, and more interpretable. This is precisely the strategy of spectral clustering.

2.2.2 Graph Laplacians

Defining graph Laplacians

Although the adjacency matrix completely describes the graph, other matrices (or "*graph operators*"), can be used for constructing spectral representations of G. One popular class of matrices used are the different graph Laplacians. The unnormalized and normalized Laplacians are of particular interest.

Definition 16. Let G be a simple graph, and denote by A its adjacency matrix. Define the *degree matrix* as the diagonal matrix D such that $D_{ii} := \sum_{ij} A_{ij}$ for each $i \in \{1, \dots, n\}$. Define the unnormalized and normalized Laplacians respectively by

$$\begin{aligned} L_{\text{unn}} &:= D - A \\ L_{\text{sym}} &:= I - D^{-1/2} A D^{-1/2}. \end{aligned} \tag{2.12}$$

There are several different ways of motivating this popularity. In what follows, this is explained from a community detection perspective, in the intuitive case of assortative communities.

The Laplacian and connectivity

Assume a disconnected graph G having k connected components. A particular instance of this is when there are k assortative communities that are completely separated. The kernel of the Laplacian contains precisely the connectivity information of the graph, and in this degenerate case this coincides with the community information. See [20] for a proof.

Proposition 2. *Let G be a simple unweighted graph with k connected components* $\Omega_1, \dots, \Omega_k$. *Then the algebraic multiplicity of the eigenvalue* 0 *of* L_{unn} *equals k and the indicator vectors* $\mathbf{1}_{\Omega_1}, \dots, \mathbf{1}_{\Omega_k}$ *span its null space.*

Remark. An analogous proposition holds for L_{sym}, with the sole difference being that it is the vectors $\{D^{1/2}\mathbf{1}_{\Omega_i}\}_{i=1,\dots,n}$ that span the null space of the Laplacian.

The Laplacian and graph cuts

Moving on from the degenerate case, consider now that the graph is perturbed and the k communities now communicate weakly, that is, that there are some edges across them. Then it makes sense to propose a method of finding the communities by seeking a partition which minimizes the number of edges crossing classes, while still keeping the partitions with a reasonable size to avoid degenerate solutions. Formalizing this yields the following definitions.

Definition 17. Let $G = (V, E)$ be a simple graph. The *cut* is a function associating any partition P of G to the number of edges connecting nodes belonging to different classes, i.e.,

$$\text{cut}(P) := \frac{1}{2} \sum_{i=1}^{n} \sum_{j : P(i) \neq P(j)} A_{ij}. \tag{2.13}$$

The first step is to normalize this metric with respect to class sizes, to avoid taking unbalanced solutions; in some cases, without normalization, the cut is minimized by separating an individual vertex from the rest of the graph, which is typically not the behavior desired for community assignments. The *ratio cut* is a possible normalization of the cut. The results that follow will associate it to the unnormalized Laplacian. Taking the alternative NCut normalization yields their analogous version for the symmetric normalized Laplacian. For the sake of simplicity, only the results using the ratio cut will be presented.

Definition 18. Let $\text{cut}(P)_{ij}$ denote the cut between classes i and j of partition P, assumed to have k classes. The ratio cut is defined as

$$\text{RatioCut}_k(P) := \frac{1}{2}\sum_{i=1}^{k}|P_i|^{-1}\sum_{\substack{j=1\\ j\neq i}}^{k}\text{cut}(P)_{ij}. \tag{2.14}$$

Definition 19. The *balanced min-cut problem* is the following optimization problem:

$$\min_{P\in\mathcal{P}_k(G)} \text{RatioCut}_k(P), \tag{2.15}$$

where $\mathcal{P}_k(G)$ denotes the set of all partitions of G into k classes.

This problem can be rewritten in terms of the Laplacian.

Proposition 3. *The balanced min-cut problem can be rewritten in terms of the Laplacian as*

$$\begin{aligned} &\min_{A_1,\dots,A_k} \text{Tr}(H^tLH)\\ &\textit{s.t. } H^tH = I, \end{aligned} \tag{2.16}$$

where $H \in \mathbb{R}^{n\times k}$ *is the matrix*

$$h_{ij} := \begin{cases} 1/\sqrt{|A_{ij}|} & \textit{if node } i \in A_j\\ 0 & \textit{otherwise.} \end{cases} \tag{2.17}$$

2.2.3 A spectral clustering algorithm

The problem in Proposition 3 is NP-hard due to its hard constraint (2.17) on the form of the matrix H. It is natural to drop this constraint to get a solvable approximation to the problem.

Definition 20. The *relaxed balanced min-cut problem* is defined by replacing the constraint (2.17) on the form of H by a more general orthogonality constraint:

$$\begin{aligned} &\min_{H \in \mathbb{R}^{n \times k}} \operatorname{Tr}(H^t L H) \\ &\text{s.t. } H^t H = I \end{aligned} \tag{2.18}$$

This kind of problem has a known solution, given by the Rayleigh-Ritz theorem. A proof is available in [9].

Proposition 4 (Rayleigh-Ritz)**.** *Problem* (2.18) *is solved by the matrix H having the first k eigenvectors of L as columns.*

Originally, H's columns indicated the communities, by constraint (2.17). One might expect that this solution to the relaxed problem might still contain this information. Therefore the final step is to go from this approximate solution to community assignments. Originally, the fact that H's columns were indicatrices for the communities means that there were only k distinct rows in it. That is, by seeing the rows of H as vectors, there were only k distinct vectors. One might expect that, seeing the rows of the approximate solution H_{approx} as vectors, these vectors still "fluctuate" around the original k distinct vectors. If this is the case, clustering these vectors might yield the community information. This clustering can be done by k-means, for example. This procedure is what is commonly called the spectral clustering method.

Remark. Spectral clustering *tries* to find a balanced minimum cut partition. It may succeed in applications, but theoretically there are simple counterexamples showing that the quality of this approximation can be arbitrarily bad, see [20]. Searching other algorithms can only partially help, as there is no general efficient algorithm for solving graph cut problems [2].

Long story short, here are the steps of the spectral clustering algorithm.

Algorithm 2 Spectral clustering

```
Require: Adjacency matrix $A$
  Calculate Laplacian $L$: $L_{\text{unn}} \leftarrow D - A$ or $L_{\text{sym}} \leftarrow I - D^{-1/2} A D^{-1/2}$
  Calculate the $k$ first eigenvectors of $L$ and put them as columns in matrix $H$
  Cluster the rows of $H$ with an algorithm such as $k$-means
  Return cluster assignments as community assignments for nodes
```

The diagram below illustrates the intuition and logic steps behind the deduction of spectral clustering algorithms.

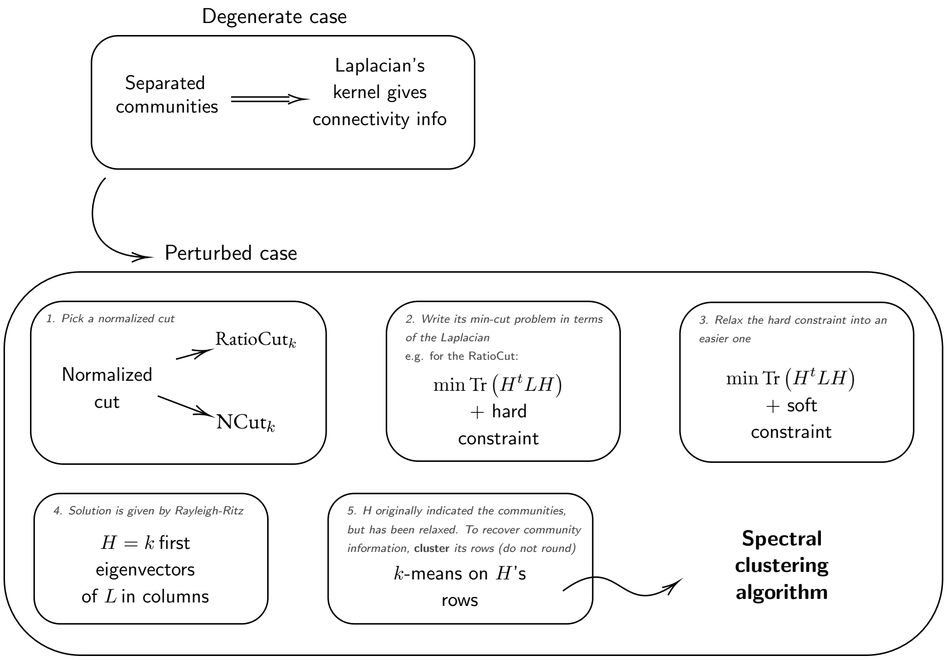

Degenerate case
Separated communities
Laplacian's kernel gives connectivity info
Perturbed case
1. Pick a normalized cut
Normalized cut
$\mathrm{RatioCut}_k$
NCut_k
2. Write its min-cut problem in terms of the Laplacian
e.g. for the RatioCut:
$\min \operatorname{Tr}\left(H^t L H\right)$ + hard constraint
3. Relax the hard constraint into an easier one
$\min \operatorname{Tr}\left(H^t L H\right)$ + soft constraint
4. Solution is given by Rayleigh-Ritz
$H = k$ first eigenvectors of L in columns
5. H originally indicated the communities, but has been relaxed. To recover community information, cluster its rows (do not round)
k-means on H's rows
Spectral clustering algorithm

Figure 2.4: Diagram describing the steps taken to derive spectral algorithms for community detection.

Chapter 3

Statistical learning theory

This chapter adresses the question of quantitatively measuring an algorithm's ability to recover *ground truth* communities on a graph sampled from a model, understanding what factors might impact its ability to perform such recovery, and then specializes this investigation to the case of the two algorithms of the previous chapter.

3.1 Agreement and degrees of recovery

This section first defines metrics for the quality of a particular estimation produced by an algorithm, and then uses these metrics to define its global ability to rexover communities.

3.1.1 Agreement

Consider the task of evaluating the answer returned by any algorithm for community detection. Assume knowledge of the true vector of community assignments $Z^\star$. Given an estimate $\hat{Z}$ for $Z^\star$ how can one measure the quality of such an estimation? The most intuitive metric for this is the *agreement*, which is simply the average number of nodes whose communities were estimated correctly, up to an arbitrary relabeling of the communities. This relabeling must be taken into account since the choice of integer associated to a community is arbitrary, and any relabeling defines the same partition of the nodes of the graph.

Definition 21 (Agreement)**.** Let $Z^\star$ and $\hat{Z}$ be, respectively, the true and an arbitrary vector of community assignments. Let also S_k denote the group of permutations of $[k]$. Define the *agreement* between $Z^\star$ and $\hat{Z}$ to be

$$\mathcal{A}(Z^\star, \hat{Z}) := \frac{1}{n} \max_{\sigma \in S_k} \sum_{i=1}^{n} \mathbf{1}(Z_i^\star = \sigma(\hat{Z}_i)). \tag{3.1}$$

When studying weaker forms of recovery, under general (asymmetric) SBMs, a *normalized* version of this metric is actually needed.

Definition 22 (Normalized agreement)**.** Let $Z^\star$ and $\hat{Z}$ be, respectively, the true and an arbitrary vector of community assignments. Let also S_k denote the group of permutations of $[k]$. Define the *normalized agreement* between $Z^\star$ and $\hat{Z}$ to be

$$\tilde{\mathcal{A}}(Z^\star, \hat{Z}) := \frac{1}{k} \max_{\sigma \in S_k} \sum_{k=1}^{K} \frac{\sum_{i=1}^{n} \mathbf{1}(Z_i^\star = \sigma(\hat{Z}_i))\mathbf{1}(Z_i^\star = k)}{\sum_{i=1}^{n} \mathbf{1}(Z_i^\star = k)}. \tag{3.2}$$

3.1.2 Degrees of recovery

Definition 21 can be used to measure different degrees of performance of algorithms whose task is to estimate the communities $Z^\star$. The degree to which an algorithm is capable of recovering the communities is captured in what are called the different (asymptotic) *degrees of recovery*. Notice these are all defined asymptotically. For a review of asymptotic notation, see Section A.6.

Definition 23 (Degrees of recovery)**.** Let $(Z^\star, G) \sim \text{SBM}(n, \pi^\star, \Gamma^\star)$, and $\hat{Z}$ be the output of an algorithm taking $(G, \pi^\star, \Gamma^\star)$ as input. Then, the following *degrees of recovery* are said to be *solved* if, asymptotically on n, one has

- Exact recovery $\leftrightarrow \mathbb{P}(\mathcal{A}(Z^\star, \hat{Z}) = 1) = 1 - o(1)$,
- Almost exact recovery $\leftrightarrow \mathbb{P}(\mathcal{A}(Z^\star, \hat{Z}) = 1 - o(1)) = 1 - o(1)$,
- Partial recovery $\leftrightarrow \mathbb{P}(\tilde{\mathcal{A}}(Z^\star, \hat{Z}) \geq \alpha) = 1 - o(1), \alpha \in (1/k, 1)$,
- Weak recovery (also called *detection*) $\leftrightarrow \mathbb{P}(\tilde{\mathcal{A}}(Z^\star, \hat{Z}) \geq 1/k + \Omega(1)) = 1 - o(1)$.

Remark. There is an intuition for why $\alpha > 1/k$ in the definitions of partial and weak recovery above. If one assumes knowledge of the true parameters $(\pi^\star, \Gamma^\star)$ of the model when designing an estimation algorithm, then the trivial algorithm of simply assigning each node a random community according to $\pi^\star$ will achieve an agreement of $\|\pi\|_2^2$, by the law of large numbers. In particular, in the case where the communities are uniform, $\pi = \mathbf{1}_k/k$ and the trivial agreement reached will be $\mathcal{A} = 1/k$.

3.2 Asymptotic topologies

This section details one factor that might impact algorithms' performance in the context of the analysis performed in this chapter: the asymptotic structure of the graphs to which they are applied to. First, it is explained how these structures appear in models relevant for this report. Then, it is explained how this impacts the robustness of algorithms.

3.2.1 The case of the Erdős-Rényi model

The first random graph model was the Erdős-Rényi model, denoted $G(n, p)$, over graphs with n vertices [4]. Under this model, the presence of an edge between each pair of nodes is determined by a Bernoulli random variable of parameter p. See Figure 3.1.

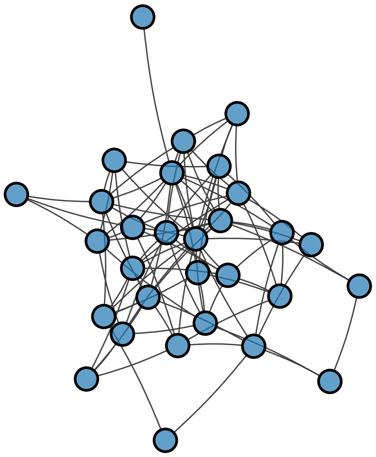


Figure 3.1: An observation from an Erdős-Rényi model $G(30, 0.2)$

Although this model does not present clusters of nodes, it is nevertheless of great interest, since it reveals a key phenomenon: there exist tightly defined and distinct *asymptotic topologies* for random graphs arising from this model as $n \to \infty$. Which one arises is a function of the growth of p with respect to n.

Theorem 2.

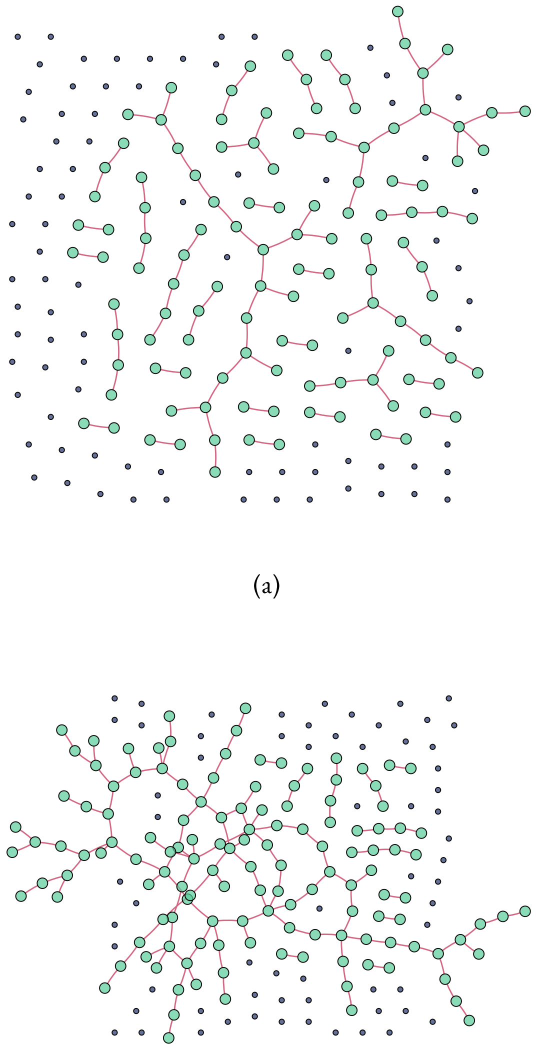
3.2.2 The case of the SBM

In the SBM, essentially the same phenomenon happens.

Proposition 5. *Content.*

3.2.3 Why does this matter?

This phenomenon is of interest for community detection, because the analysis of algorithms takes place asymptotically. Knowing what structures appear can provide an intuition to the common hypotheses appearing on consistency results regarding the asymptotic regime of the expected degree. It can also reveal weaknesses of some methods that rely too heavily on these structures, implying a lack of robustness to perturbations.



(a)

(b)

Figure 3.2: (a) At $np = 0.8 < 1$, there are some small trees of size at most $O(\log(n))$. (b) At $np = 1.33 > 1$ a giant component appears, of size $O(n^{2/3})$.

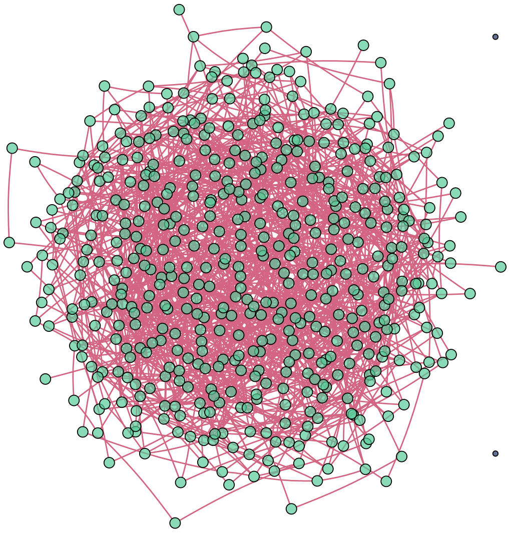


Figure 3.3: At $p = 0.011 < 0.012$ there exists almost surely an isolated vertex, and the graph is disconnected. When $p = 0.013 > 0.012$, isolated vertices disappear almost surely, and the graph finally becomes connected.

, and this affects directly the performance of algorithms, and has led to ideas of how to increase their robustness.

3.3 Consistency of variational EM

The main idea is that in the sum over all latent variables that defines the likelihood, this sum actually concentrates on a few important terms that we can identify.

3.4 The case of of spectral clustering

3.4.1 Motivation

The work of [17] studies the question of whether spectral clustering is capable of recovering the communities of graphs generated under an SBM. Although they do not imply consistency of the algorithm and have quite strong hypotheses, their results became very popular for the intuitions they give. The objective of this section is to explain them and to understand a bit better when and *why* the spectral clustering algorithm might work or not on graphs generated from an SBM. Some details will be given in the appendices, but the main objective is to make the ideas clear.

3.4.2 Notations

The matrix $L = D^{-1/2}AD^{-1/2}$ is used as Laplacian. At first sight, it may seem different from the normalized Laplacian L_{sym} defined in (2.12). However, one can directly check these matrices have the same eigenvectors: if $L_{\text{sym}}v = \lambda v$, then $Lv = (1-\lambda)v$, and if $Lv = \lambda v$, then $L_{\text{sym}}v = (1-\lambda)v$. The difference of eigenvalues simply implies that instead of taking the k smallest eigenvectors one must take the k largest; this being taken into account, these matrices are equivalent for the analysis.

Remark. Do pay attention when translating these results to the unnormalized Laplacian, since its eigenvectors *will not* be equal to those of L (but they can easily be transformed using $D^{1/2}$).

The *expected* adjacency matrix is defined as $\mathscr{A} := \mathbb{E}[A|Z^\star]$, and the *expected* degree matrix is the diagonal matrix with entries $\mathscr{D}_{ii} = \sum_k \mathscr{A}_{ik}$. Naturally, the expected Laplacian is defined as $\mathscr{L} = \mathscr{D}^{-1/2}\mathscr{A}\mathscr{D}^{-1/2}$. A quantity that will play a fundamental role is defined as $\rho_n := \min_{i=1,\dots,n} \mathscr{D}_{ii}^{(n)}/n$. Intuitively, it measures how quickly the number of edges accumulates as the number of nodes in the graphs generated grows. Notice that D_{ii} is the *expected degree* of community i.

The structure of the results is diagrammatically shown in Figure 3.4, and the analysis is divided in two parts. First, consider the question of whether the empirical counterpart of the expected Laplacian approaches it as the size of the graphs observed grows. This could be hoped for, since in the case of the adjacency matrix, the entries of $A|Z$ are independent (technically, its upper triangular part is independent, and the lower triangular part is identical to it), thus there is a strong concentration of A around its expected version. As it will be seen, it is not as straightforward for the Laplacian, but fortunately with some additional work it is possible to show a similar concentration for it. Second, there is the question of whether the expected Laplacian contains or not the information on the communities. If it does, and given that the empirical Laplacian approaches it, then one could try to estimate this information from the empirical version.

3.4.3 Convergence of eigenvectors

The first part consists in showing a preliminary result of concentration of the empirical Laplacian towards its expectation, that is, showing that the eigenvectors of the empirical Laplacian converge in some sense to the eigenvectors of the expected Laplacian. This conclusion would be immediate if the former converged to the latter in Frobenius norm, i.e. if $\|L^{(n)} - \mathscr{L}^{(n)}\|_F \to 0$, since then the *Davis-Kahan* theorem [21] would imply the alignment of their eigenspaces. Unfortunately, these matrices do not converge in Frobenius norm [17], so a "detour" is needed in order to achieve this convergence.

What is demonstrated is that, under certain conditions, their "squares" converge in Frobenius norm.

Proposition 6. *If there exists some $N > 0$ such that $\rho_n^2 \log n > 2$ for all $n > N$, then*

$$\|L^{(n)}L^{(n)} - \mathscr{L}^{(n)}\mathscr{L}^{(n)}\|_F = o\left(\frac{\log n}{\rho_n^2 n^{1/2}}\right) \quad a.s. \tag{3.3}$$

It is important to interpret the hypothesis on ρ_n. It is a type of hypothesis commonly present in consistency results such as this one. It sets the growth regime for the expected degree in order for the result to hold. As seen in 3.2, this type of hypothesis is related to the asymptotic topology of the graphs arising from the model. In a sense, the strictness imposed by this hypothesis can be used to measure the "strength" of a given consistency result. The stronger this hypothesis, the denser the graphs must be for the result to hold, and the less robust the proof will be to sparsity.

A version of the Davis-Kahan theorem then implies that, under some conditions, the eigenvectors of $L^{(n)}L^{(n)}$ converge to the eigenvectors of $\mathscr{L}^{(n)}\mathscr{L}^{(n)}$, up to some (unknown) *rotation*. This in turn can be transformed into a convergence statement for the eigenvectors of the Laplacian without the square. The precise statement of the theorem is quite long (due to an hypothesis dealing with the rate with which the eigengap closes; the eigengap is the minimal distance between an eigenvalue of the eigenvectors being used in the algorithm, and the rest of the eigenvalues); see [17] for the details and a complete statement.

Proposition 7. *If there exists $N > 0$ such that $\rho_n^2 \log n > 2$ for all $n > N$ and if the gap δ_n between the eigenvalues of interest and the other eigenvalues does not go to zero too*

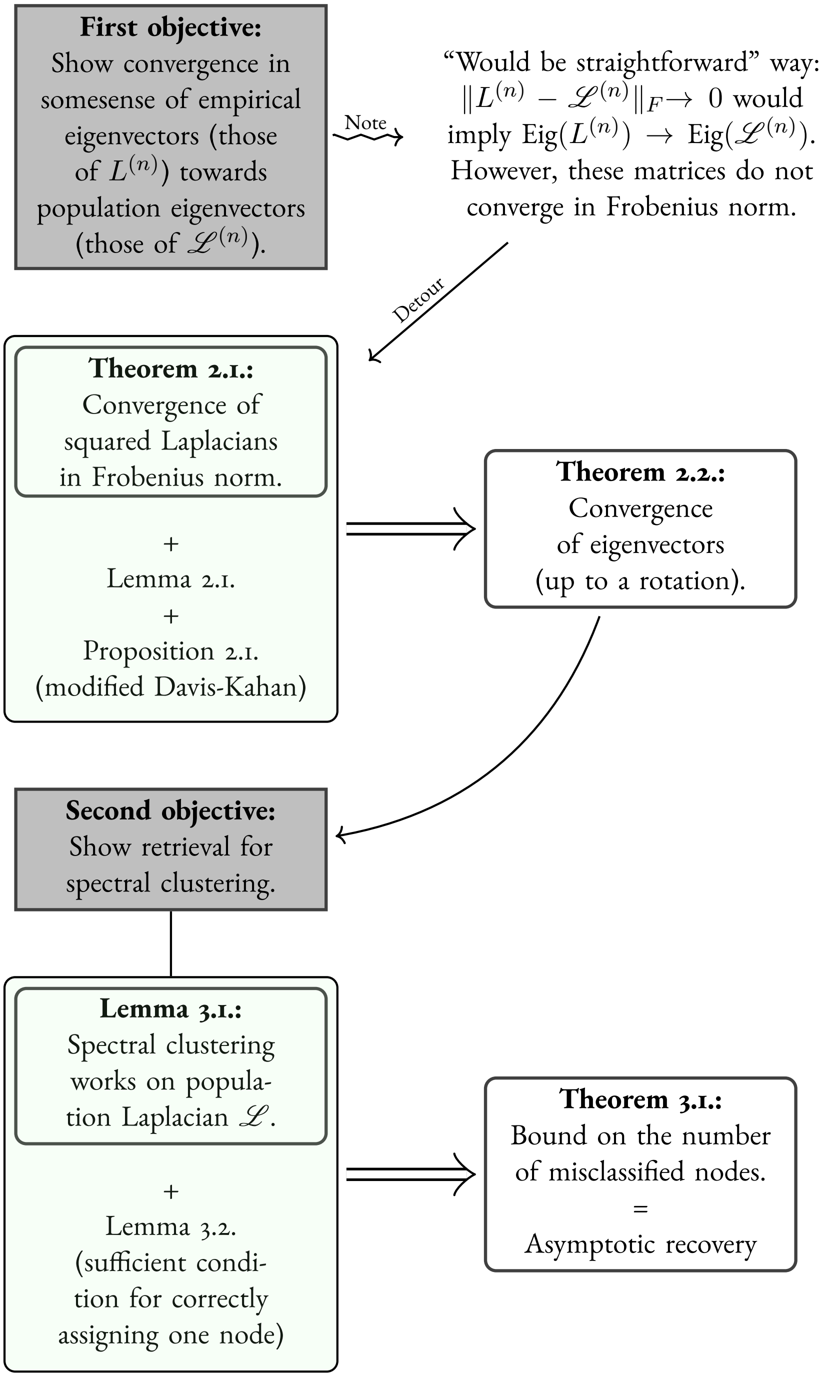

First objective:
Show convergence in somesense of empirical eigenvectors (those of $L^{(n)}$) towards population eigenvectors (those of $\mathscr{L}^{(n)}$).
Note
"Would be straightforward" way: $\|L^{(n)} - \mathscr{L}^{(n)}\|_F \to 0$ would imply $\mathrm{Eig}(L^{(n)}) \to \mathrm{Eig}(\mathscr{L}^{(n)})$. However, these matrices do not converge in Frobenius norm.
Detour
Theorem 2.1.:
Convergence of squared Laplacians in Frobenius norm.
+
Lemma 2.1.
+
Proposition 2.1.
(modified Davis-Kahan)
Theorem 2.2.:
Convergence of eigenvectors (up to a rotation).
Second objective:
Show retrieval for spectral clustering.
Lemma 3.1.:
Spectral clustering works on population Laplacian $\mathscr{L}$.
+
Lemma 3.2.
(sufficient condition for correctly assigning one node)
Theorem 3.1.:
Bound on the number of misclassified nodes.
=
Asymptotic recovery

Figure 3.4: Diagram of results in [17]. First, the question of whether the empirical Laplacians approach the expected version or not is studied; then, the question of whether the expected Laplacian contains or not the complete information on the communities is analyzed.

quickly (cf. [17]), then for some sequence of orthonormal rotations O_n,

$$\|X_n - \mathscr{X}_n O_n\|_F = o\left(\frac{\log n}{\delta_n \rho_n^2 n^{1/2}}\right) \quad a.s., \tag{3.4}$$

where X_n is the orthogonal matrix with the eigenvectors of $L^{(n)}$ as columns; similarly for $\mathscr{X}_n$ and $\mathscr{L}^{(n)}$.

In short, the convergence of eigenvectors takes place up to a rotation. This rotation is immaterial for the spectral clustering algorithm, since the k-means will cluster the empirical eigenvectors solely based on their relative concentration.

3.4.4 Bounding the number of misclassified nodes

The second part of the analysis is showing that the estimation $\hat{Z}^{(\mathrm{Alg})}_{\mathrm{Spec}}$ obtained using the spectral clustering algorithm asymptotically matches the ground truth assignments $Z^\star$. It starts by showing that applying the algorithm to the expected Laplacian (instead of the observed one) recovers the partitions of the SBM, even non-asymptotically.

Lemma 1. *There exists a matrix $\mu \in \mathbb{R}^{k\times k}$ such that the eigenvectors of $\mathscr{L}$ corresponding to the nonzero eigenvalues are the columns of $Z\mu$. Furthermore, it holds that*

$$z_i\mu = z_j\mu \iff z_i = z_j, \tag{3.5}$$

where z_i is the i-th row of Z.

Proof. A sketch of the proof is given in Section A.5. □

The meaning of equivalence (3.5) is that rows i and j of $Z\mu$ are equal if, and only if, the corresponding rows of Z are equal, that is if nodes i and j belong to the same community. Since there are k communities, this implies that there can be at most k unique rows in the matrix $Z\mu$ of eigenvectors of $\mathscr{L}$. Spectral clustering applies k-means to these vectors, and thus these become precisely the centroids of k-means (since one is applying k-means to at most k different vectors). The rows of $Z\mu$ will then be attributed to the centroid they are equal to, and by the equivalence in Equation (3.5), this implies that spectral clustering applied to the expected Laplacian $\mathscr{L}$ identifies the communities. The rows $z_i\mu$ constitute what will be called the *expected* centroids. Of course, the above is a theoretical result, since in practice the expected Laplacian is unknown.

To go from this theoretical consideration to an empirical one, consider that k-means is applied to the rows of X, the matrix whose columns are the first k eigenvectors of the empirical Laplacian L. For node i, denote c_i the *empirical* centroid associated to it by the k-means algorithm. Intuitively, it is expected that if the algorithm works, then the empirical centroid c_i is closer to the expected centroid $z_i\mu$ than to the other $z_j\mu, j \neq i$ expected centroids, since these represent communities distinct of that of i. However, remember that a rotation O needed to be included in proposition 7, and it must also be included here. The main result in [17] bounds the size of the set $\mathscr{M}$ of nodes i closer to some (rotated) expected centroid $Oz_j\mu, j \neq i$ associated to another community than to its rotated expected centroid $Oz_i\mu$. [17] call the nodes in $\mathscr{M}$ "misclassified" nodes, although this is misleading since it does not make any reference to the estimated labels nor to the ground truth communities.

Theorem 3. *Define $P_n := \max_{j=1,\dots,k}(Z^tZ)_{jj}$, which is the size of the largest community for graph-sample n. Denote by λ_n the smallest non-zero eigenvalue of $\mathscr{L}_n$. If $n^{-1/2}(\log n)^2 = O(\lambda_n^2)$ and there exists N such that for all $n > N, \rho_n^2 > 2/\log n$, then*

$$|\mathscr{M}| = o\left(\frac{P_n(\log n)^2}{\lambda_n^4\rho_n^4 n}\right). \tag{3.6}$$

Intuitively, this result states that if the smallest eigenvalue of the expected Laplacian does not decrease too fast, and if the expected degree grows fast enough, then the empirical centroids c_i will be closer the "correct" rotated expected centroids $Oz_i\mu$ than to the other $Oz_j\mu$, with a proportion tending to one. Notice however that this result does not imply convergence of centroids to their expected version, and more importantly, *does not* show consistency of spectral clustering. It is important to also emphasize that the conditions imposed by Theorem 3 are quite strong. For instance, the condition on the degree, $\rho_n^2 \log n > 2$ is almost as strong as requiring that the expected degrees grow as n. That is, very dense graphs are required, and many real graphs are sparser than this, in a way one cannot expect these results to hold in practice.

Chapter 4

Numerical experiments

4.1 Numerical experiments

Chapter 5

Conclusion and outlook

5.1 Conclusions

Your text.

Appendix A

Proofs, calculations, extra definitions

A.1 Proof of Theorem 1

The posterior is defined as

$$p\left(Z|A;\theta\right)=\frac{p\left(Z,A;\theta\right)}{p\left(A;\theta\right)},$$

which implies

$$p\left(A;\theta\right)=\frac{p\left(Z,A;\theta\right)}{p\left(Z|A;\theta\right)}.$$

Thus,

$$\log p\left(A;\theta\right)=\log p\left(Z,A;\theta\right)-\log p\left(Z|A;\theta\right).$$

Taking the expectation of this expression with respect to some proposal distribution $q(Z)$ depending only on Z, one has

$$\begin{aligned}\log p\left(A;\theta\right)&=\int_Z \log p\left(Z,A;\theta\right)q\left(Z\right)\,dZ-\int_Z \log p\left(Z|A;\theta\right)q\left(Z\right)\,dZ\\&=\int_Z\left(\log\left(\frac{p\left(Z,A;\theta\right)}{q\left(Z\right)}\right)-\log q\left(Z\right)\right)q\left(Z\right)\,dZ\\&\quad-\int_Z\left(\log\left(\frac{p\left(Z|A;\theta\right)}{q\left(Z\right)}\right)-\log q\left(Z\right)\right)q\left(Z\right)\,dZ.\end{aligned}$$

Therefore,

$$\log p\left(A;\theta\right)=\int_Z\log\left(\frac{p\left(Z,A;\theta\right)}{q(Z)}\right)q(Z)\,dZ-\int_Z\log\left(\frac{p\left(Z|A;\theta\right)}{q(Z)}\right)q(z)\,dZ,$$

that is,

$$\log p\left(A;\theta\right)=F\left(q,\theta\right)+\mathrm{KL}\left(q\|p\left(Z|A;\theta\right)\right).$$

A.2 Proof of Proposition 1

Similarly to [10], substitute Equation (2.7) into the ELBO of Equation (2.5), to get

$$\begin{aligned}F=\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{E}_q\left[Z_{ik}\right]\log\pi_k+\sum_{j>i}A_{ij}\mathbb{E}_q\left[\log\gamma_{Z_i,Z_j}\right]\\+\left(1-A_{ij}\right)\mathbb{E}_q\left[\log\left(1-\gamma_{Z_i,Z_j}\right)\right]+\mathcal{H}\left(q\right),\end{aligned}\tag{A.1}$$

where $\mathcal{H}(q)=-\sum_Z q(Z)\log(Z)$ is the entropy of the distribution q. This expression can be further simplified by noticing that the expectations appearing in it can be explicitly calculated. First, notice that $\mathbb{E}_q[Z_{ik}]=\tau_{ik}$. Also notice that for each i,j,

$$\begin{aligned}\mathbb{E}_q\left[\log\gamma_{Z_i,Z_j}\right]&=\sum_{Z}q\left(Z\right)\log\gamma_{Z_i,Z_j}\\&=\sum_{Z_i,Z_j}q\left(Z_i,Z_j\right)\log\gamma_{Z_i,Z_j}\\&=\sum_{Z_i,Z_j}m\left(Z_i,\tau_i\right)m\left(Z_j,\tau_j\right)\log\gamma_{Z_i,Z_j}\\&=\sum_{Z_i,Z_j}\prod_{k=1}^{K}\tau_{ik}^{Z_{ik}}\prod_{l=1}^{K}\tau_{jl}^{Z_{jl}}\log\gamma_{Z_i,Z_j}\\&=\sum_{k=1}^{K}\sum_{l=1}^{K}\tau_{ik}\tau_{jl}\log\gamma_{kl}.\end{aligned}$$

Similarly, $\mathbb{E}_q[\log(1-\gamma_{Z_i,Z_j})] = \sum_{k,l}\tau_{ik}\tau_{jl}\log(1-\gamma_{kl})$. The entropy term can also be simplified by noticing that

$$\begin{aligned}
\mathcal{H}(q) &= \mathcal{H}\left(\prod_{i=1}^{n} m(Z_i;\tau_i)\right) \\
&= -\sum_{Z}\left(\prod_{i=1}^{n} m(Z_i;\tau_i)\right)\log\prod_{j} m(Z_j;\tau_j) \\
&= -\sum_{j=1}^{n}\sum_{Z}\left(\prod_{i=1}^{n} m(Z_i;\tau_i)\right)\log m(Z_j;\tau_j) \\
&= -\sum_{j=1}^{n}\sum_{Z_j} m(Z_j,\tau_j)\log m(Z_j,\tau_j) \\
&= \sum_{j=1}^{n}\mathcal{H}(m(Z_j,\tau_j)),
\end{aligned}$$

where the sum in Z became a sum in Z_j by marginalizing out the remaining multinomial mass functions. Each such term can be calculated:

$$\begin{aligned}
\mathcal{H}(m(Z_j,\tau_j)) &= -\sum_{Z_j} m(Z_j,\tau_j)\log m(Z_j,\tau_j) \\
&= -\sum_{Z_j}\left(\prod_{i=1}^{K}\tau_{ji}^{Z_{ji}}\right)\sum_{k=1}^{K} Z_{jk}\log\tau_{jk} \\
&= -\sum_{i=1}^{K}\tau_{ji}\log\tau_{ji},
\end{aligned}$$

where the expression is simplified by considering that the possible values for Z_j are the different indices where its only non-zero component can be. Substituting all of this back into Equation (A.1), one finds that the expression for the mean-field ELBO, as in the theorem statement.

A.3 Proof of EM explicit steps

Having fixed a value for $\hat{\theta}$, maximize the mean-field ELBO with respect to τ under the constraint that $\tau_{ik} \geq 0$ for $1 \leq i \leq n$ and $1 \leq k \leq K$, and that $\sum_k \tau_{ik} = 1$ for $1 \leq i \leq n$; then, for a fixed $\hat{\tau}$, maximize the mean-field ELBO with respect to θ under the constraint $0 < \pi_k < 1, 1 \leq k \leq K$ and so on.

Optimizing for $\hat{\tau}$

By penalizing the constraints for τ on the ELBO, one gets a Lagrangian $\mathscr{L}$ which when derived and equaled to zero yields

$$\begin{aligned}\nabla_{\tau_{ik}} \mathscr{L} = \log \pi_k - 1 - \log \tau_{ik} + & \sum_{\substack{j>i \\ l=1,\ldots,K}} A_{ij} \tau_{jl} \log \gamma_{kl} \\ & + (1 - A_{ij}) \tau_{jl} \log (1 - \gamma_{kl}) + \mu_i \\ = 0. &\end{aligned}$$

This can be directly rearranged to the following fixed point relation

$$\hat{\tau}_{ik} \propto \pi_k \prod_{\substack{j>i \\ l=1,\ldots,K}} \left(\gamma_{kl}^{A_{ij}} \left(1 - \gamma_{kl}\right)^{(1-A_{ij})} \right)^{\tau_{jl}}. \tag{A.2}$$

Thus, one "general way" of obtaining $\hat{\tau}$ is by evaluating Equation (A.2) repeatedly until convergence.

Optimizing for $\hat{\theta}$

As π_k appears inside a logarithm in the ELBO, the positivity constraint is naturally enforced by the objective function. Therefore one only needs to impose that it sums to one. The Lagrangian then becomes

$$\mathscr{L} = -\sum_{i=1}^{n} \sum_{k=1}^{K} \tau_{ik} \log \frac{\pi_k}{\tau_{ik}} + \mu \left(\sum_{l=1}^{K} \pi_l - 1 \right),$$

and its derivative equaled to zero provides the equation

$$\nabla_{\pi_k} \mathscr{L} = -\sum_{i=1}^{n} \frac{\tau_{ik}}{\pi_k} + \mu = 0.$$

This optimality condition gives the estimator sought in terms of the Lagrange multiplier

$$\hat{\pi}_k = \frac{1}{\mu} \sum_{i=1}^{n} \tau_{ik}.$$

To find the value of the Lagrange multiplier, one must solve the dual problem. Lagrange's dual function writes

$$q(\mu) = -\sum_{i=1}^{n} \sum_{k=1} \tau_{ik} \log \left(\frac{\sum_{l=1}^{K} \tau_{lk}}{\mu \tau_{ik}} \right) + \tau_{ik} - \mu.$$

The dual problem is $\max_\mu q(\mu)$, which is unconstrained:

$$\nabla_\mu q = \sum_{i=1}^{n}\sum_{k=1}^{K}\frac{\tau_{ik}}{\mu} - 1 = 0 \implies \mu = \sum_{i=1}^{n}\sum_{k=1}^{K}\tau_{ik} = n.$$

Substituting the value found for μ back into the estimator for $\hat{\pi}$, one concludes

$$\hat{\pi}_k = \frac{1}{n}\sum_{i=1}^{n}\tau_{ik}. \tag{A.3}$$

A similar procedure generates estimators for the connectivities γ, the difference being that these are unconstrained, so that by deriving the ELBO directly one gets

$$\begin{aligned}\nabla_{\gamma_{kl}}\mathcal{L} = \sum_{i=1}^{n}\sum_{j=1}^{n}\tau_{ik}\tau_{jl}\left(\frac{A_{ij}}{\gamma_{kl}} - \frac{1-\delta_{ij}-A_{ij}}{1-\gamma_{kl}}\right) = 0 \\ \implies \sum_{i=1}^{n}\sum_{j=1}^{n}\tau_{ik}\tau_{jl}\left(A_{ij} - \gamma_{kl}\left(1-\delta_{ij}\right)\right) = 0.\end{aligned}$$

This optimality condition yields the estimator sought:

$$\hat{\gamma}_{kl} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\tau_{ik}\tau_{jl}A_{ij}}{\sum_{i=1}^{n}\sum_{j=1}^{n}\tau_{ik}\tau_{jl}\left(1-\delta_{ij}\right)}. \tag{A.4}$$

These equations can be simplified in the simple case of two communities, which is a fundamental example for building intuition and testing hypotheses.

A.4 Convergence of VEM to a local maximum

Showing that the algorithm increases the likelihood at each step (and therefore achieves some local maximum) is very similar to the case of the classical EM.

Proposition 8. *The VEM algorithm increases the likelihood at each step.*

Proof. For any fixed θ_0, the variational decomposition is

$$\log p(x;\theta_0) = F(q_\tau, \theta_0) + \mathrm{KL}(q_\tau \| p(\cdot|x;\theta_0)). \tag{A.5}$$

Observe that in general the KL term can no longer be zero, but it can be minimized, leading to the best approximation to the posterior within the mean-field family. Minimizing the KL still is equivalent to maximizing the ELBO, thus the new *variational* E step consists in finding τ, fixed θ_0:

$$\tau_0 = \operatorname*{argmax}_{\tau} F(q_\tau, \theta_0). \tag{A.6}$$

Now fix τ_0 and consider a general θ in the variational decomposition:

$$\log p(x;\theta) = F(q_{\tau_0},\theta) + \mathrm{KL}(q_{\tau_0}\|p(\cdot|x;\theta)).$$

If you take $\theta_1 \coloneqq \mathrm{argmax}_\theta\, F(q_{\tau_0},\theta)$, then

$$\begin{aligned}\log p(x;\theta_1) &= F(q_{\tau_0},\theta_1) + \mathrm{KL}(q_{\tau_0}\|p(\cdot|x;\theta_1))\\ &\geq F(q_{\tau_0},\theta_0) + \mathrm{KL}(q_{\tau_0}\|p(\cdot|x;\theta_0))\\ &= \log p(x;\theta_0).\end{aligned}$$

That is, this choice of a next θ makes the observed log-likelihood grow. This maximization is the variational analogue of the M step. Notice that the observed log-likelihood does not depend on the q_τ chosen, thus the M-step keeps the observed log-likelihood constant, while the E-step makes it grow, and so overall it must grow after each EM alternation. Notice that the ELBO itself grows in both steps. □

However, there can be multiple uninformative local maxima, and it is known [18] that in closely related algorithms these bad optima can attract almost all initializations for the parameters. Consider then the question whether this algorithm converges to the *true* global maximum as $n \to \infty$, that is, the question concerning its asymptotic consistency.

A.5 Sketch of proof of Lemma 1

Here is an outline of the steps taken in the proof.

1. Factor $\mathscr{L}$ as $\mathscr{L} = Z\Gamma_L Z^t$ for some matrix $\Gamma_L \in \mathbb{R}^{k\times k}$ that can explicitly be found.

2. Consider now the (different) matrix $(Z^tZ)^{1/2}\Gamma_L(Z^tZ)^{1/2}$. It is the decomposition given for $\mathscr{L}$ under the change $Z \to (Z^tZ)^{1/2}$, which can be thought of as a "square matrix version" of Z.

3. Show that $(Z^tZ)^{1/2}\Gamma_L(Z^tZ)^{1/2}$ is symmetric and positive-definite, implying the spectral decomposition $(Z^tZ)^{1/2}\Gamma_L(Z^tZ)^{1/2} = V\Lambda V^t$.

4. Multiply the spectral decomposition on both sides by $(Z^tZ)^{-1/2}Z^t$, revealing that
$$Z\Gamma_L Z^t = \mathscr{L} = (Z\mu)\Lambda(Z\mu)^t \qquad \text{(A.7)}$$
for $\mu \coloneqq (Z^tZ)^{-1/2}V$.

5. Together with the fact that $(Z\mu)^t(Z\mu) = I_k$, where I_k is the $k \times k$ identity, Equation A.7 is precisely the eigenvector equation for $\mathscr{L}$. This shows that the columns of $Z\mu$ are the eigenvectors of $\mathscr{L}$ associated to the non-zero eigenvalues.

6. Finally, the equivalence is a direct consequence of the fact that μ is invertible:

$$\det(\mu) = \det((Z^tZ)^{-1/2})\det(V) > 0.$$

A.6 Asymptotic notation

This section some asymptotic notations for sequences that are commonly used in probability. To avoid redundancy, in what follows, let $(x_n)_{n\in\mathbb{N}}$ and $(y_n)_{n\in\mathbb{N}}$ be a pair of real sequences.

Definition 24. One denotes $x_n = O(y_n)$ (read x_n *is big-oh* y_n) if there exists some $N \in \mathbb{N}$ and some real constant $C > 0$ such that $|x_i| \leq C|y_i|$ for all $i > N$.

Definition 25. One denotes $x_n = o(y_n)$ (read x_n *is little-oh* y_n) if for every $C > 0$ there exists some $N \in \mathbb{N}$ such that for all $n > N$, $|x_n| < C|y_n|$.

Definition 26. One denotes $x_n = \Omega(y_n)$ (read x_n *is capital-omega* y_n) if $y_n = O(x_n)$.

Definition 27. One denotes $x_n = \omega(y_n)$ (read x_n *is little-omega* y_n) if $y_n = o(x_n)$.

Appendix B

The case of two communities

B.1 The case of two communities*

B.1.1 Rewriting the ELBO

Suppose now that $k = 2$. Then, for any node of index i, $\tau_{i2} = 1 - \tau_{i1}$. Thus one can work only with the first component τ_{i1}, which will be denoted from now on by τ_i. A similar logic applies to π since $\pi_2 = 1 - \pi_1$, thus subsequently work only with π_1 which will be denoted simply π. The ELBO from 2.8 writes

$$\begin{aligned}\mathcal{L}_{(k=2)} = \sum_{i=1}^{n} \tau_i \log \frac{\pi}{\tau_i} + (1-\tau_i) \log \left(\frac{1-\pi}{1-\tau_i} \right) + \frac{1}{2} \sum_{j \neq i} A_{ij} \left(\tau_i \tau_j \log \gamma_{11} \right. \\ \left. + \tau_i (1-\tau_j) \log \gamma_{12} + (1-\tau_i) \tau_j \log \gamma_{21} + (1-\tau_i)(1-\tau_j) \log \gamma_{22} \right) \\ + \frac{1}{2} \sum_{j \neq i} (1-A_{ij}) \left(\tau_i \tau_j \log (1-\gamma_{11}) + \tau_i (1-\tau_j) \log (1-\gamma_{12}) \right. \\ \left. + (1-\tau_i) \tau_j \log (1-\gamma_{21}) + (1-\tau_i)(1-\tau_j) \log (1-\gamma_{22}) \right).\end{aligned}$$

This expression can be grouped differently:

$$\begin{aligned}\mathcal{L}_{(k=2)} = \sum_{i=1}^{n} \tau_i \log \frac{\pi}{\tau_i} + (1-\tau_i) \log \left(\frac{1-\pi}{1-\tau_i} \right) \\ + \frac{1}{2} \sum_{j \neq i} \tau_i \tau_j \left(A_{ij} \log \gamma_{11} + (1-A_{ij}) \log (1-\gamma_{11}) \right) \\ + \sum_{j \neq i} \tau_i (1-\tau_j) \left(A_{ij} \log \gamma_{12} + (1-A_{ij}) \log (1-\gamma_{12}) \right) \\ + \frac{1}{2} \sum_{j \neq i} (1-\tau_i)(1-\tau_j) \left(A_{ij} \log \gamma_{22} + (1-A_{ij}) \log (1-\gamma_{22}) \right).\end{aligned}$$

To make things simpler, write this in matrix notation. Let $\mathbf{1}_n := (1, \ldots, 1)$ be a vector of dimension n, I_n be the identity matrix of size $n \times n$, and $J := \mathbf{1}_n \mathbf{1}_n^t - I_n$ be the matrix with zeros on the diagonal and ones everywhere else. The previous expression for the ELBO becomes

$$\begin{aligned}\mathcal{L}_{(k=2)} = &\sum_{i=1}^{n} \tau_i \log \frac{\pi}{\tau_i} + (1-\tau_i) \log \left(\frac{1-\pi}{1-\tau_i} \right) \\ &+ \frac{1}{2} \left(\tau^t A \tau \log \gamma_{11} + \tau^t (J - A) \tau \log (1 - \gamma_{11}) \right) \\ &+ \left(\tau^t A (\mathbf{1}_n - \tau) \log \gamma_{12} + \tau^t (J - A) (\mathbf{1}_n - \tau) \log (1 - \gamma_{12}) \right) \\ &+ \frac{1}{2} \left((\mathbf{1}_n - \tau)^t A (\mathbf{1}_n - \tau) \log \gamma_{22} \right. \\ &\qquad\qquad \left. + (\mathbf{1}_n - \tau)^t (J - A) (\mathbf{1}_n - \tau) \log (1 - \gamma_{22}) \right). \end{aligned} \tag{B.1}$$

B.1.2 The Φ function

When one observes a graph from an SBM, it is typically not the case that any parameter of the model is known, thus it is natural to consider the function $\Phi(\tau) := \sup_{\pi,\gamma} \mathcal{L}(\tau; \pi, \gamma)$. It is then natural to consider its empirical version by substituting the parameters by their estimators found in equations A.3 and 2.11. Notice that in the binary case, the expression in A.3 becomes

$$\hat{\pi} = \frac{\mathbf{1}_n^t \tau}{n}, \tag{B.2}$$

while the expression in 2.11 becomes

$$\hat{\gamma}_{11} = \frac{\tau^t A \tau}{\tau^t J \tau}, \tag{B.3}$$

$$\hat{\gamma}_{12} = \hat{\gamma}_{21} = \frac{(\mathbf{1}_n - \tau)^t A \tau}{(\mathbf{1}_n - \tau)^t J \tau}, \tag{B.4}$$

$$\hat{\gamma}_{22} = \frac{(\mathbf{1}_n - \tau)^t A (\mathbf{1}_n - \tau)}{(\mathbf{1}_n - \tau)^t J (\mathbf{1}_n - \tau)}. \tag{B.5}$$

Notice also that the equation $\hat{\gamma}_{21} = \hat{\gamma}_{12}$ comes from the symmetry of A and J. Substituting these estimators in the expression for the ELBO in order to find an expression for

$\hat{\Phi}(\tau)$, one obtains the following equation after some straightforward calculations:

$$\begin{aligned}\hat{\Phi}(\tau) = &\sum_{i=1}^{n} H(\tau_i) - nH\left(\frac{\mathbf{1}_n^t \tau}{n}\right) \\ &- \frac{\tau^t J \tau}{2} H\left(\frac{\tau^t A \tau}{\tau^t J \tau}\right) \\ &- \tau^t J(\mathbf{1}_n - \tau) H\left(\frac{\tau^t A(\mathbf{1}_n - \tau)}{\tau^t J(\mathbf{1}_n - \tau)}\right) \\ &- \frac{(\mathbf{1}_n - \tau)^t J(\mathbf{1}_n - \tau)}{2} H\left(\frac{(\mathbf{1}_n - \tau)^t A(\mathbf{1}_n - \tau)}{(\mathbf{1}_n - \tau)^t J(\mathbf{1}_n - \tau)}\right),\end{aligned} \tag{B.6}$$

where $H(x) := -x \log x - (1-x)\log(1-x)$ is the entropy of a Bernoulli. Notice that for τ lying on the corners of the hypercube this corresponds to a "sure" assignment of the nodes of the graph to one of the two communities. Of course, trying to optimize such a function on the corners of the cube is NP-hard. To simplify this expression, introduce the notations

$$E_\tau := \frac{\tau^t A \tau}{2} \qquad E_{\bar{\tau}} := \frac{\bar{\tau}^t A \bar{\tau}}{2} \qquad E_M := \tau^t A \bar{\tau} \tag{B.7}$$

$$C_\tau := \frac{\tau^t J \tau}{2} \qquad C_{\bar{\tau}} := \frac{\bar{\tau}^t J \bar{\tau}}{2} \qquad C_M := \tau^t J \bar{\tau}. \tag{B.8}$$

This is motivated by the fact that in the particular case of τ lying on the vertices of the cube these quantities equal the number of edges in the community determined by τ (the nodes i such that $\tau_i = 1$) and the number of edges that there would be in the complete graph determined by these same nodes (likewise for $\bar{\tau} := \mathbf{1}_n - \tau$). The edges and would-be edges between different communities are included in this notation by dropping the $1/2$ factor. A last (simple) piece of notation is $n_\tau := \mathbf{1}_n^t \tau$, the number of nodes in the community of nodes with $\tau_i = 1$. Using these notations, the objective in B.6 can be more elegantly expressed as

$$\begin{aligned}-\frac{\hat{\Phi}(\tau)}{C} = &-\frac{1}{C}\sum_{i=1}^{n} H(\tau_i) + \frac{n}{C} H\left(\frac{n_\tau}{n}\right) \\ &+ \frac{C_\tau}{C} H\left(\frac{E_\tau}{C_\tau}\right) + \frac{C_M}{C} H\left(\frac{E_M}{C_M}\right) + \frac{C_{\bar{\tau}}}{C} H\left(\frac{E_{\bar{\tau}}}{C_{\bar{\tau}}}\right).\end{aligned} \tag{B.9}$$

The minus sign is introduced so that this becomes an objective function to minimize, as is standard in optimization. The division by $C := \binom{n}{2}$ is for normalization purposes. This is a non-convex function on τ, which complicates its optimization.

B.1.3 Expected ELBO as objective function

Direct maximization of the $\hat{\Phi}$ function can be challenging, as it is not convex, and its solution might be in the interior of the hypercube. However, it is reasonable to expect that the maximum of the ELBO $\mathcal{L}$ should converge to the maximum of the expected ELBO $\mathbb{E}[\mathcal{L}|Z]$, where the expectation is taken with respect to the randomness of A and assuming knowledge of the model parameters and Z. The expected ELBO should be simpler to treat, and intuitively its maximum should be the assignments Z. Numerical simulation supports this intuition.

Thus, proceed in two steps. First, show that the maximum of the expected ELBO is indeed the vector of assignments Z. This will be done first on the simple case of the SBM with two communities. Then, properly show the convergence of the maxima of the ELBO to Z.

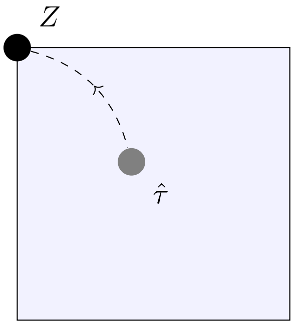


Figure B.1: The maximum $\hat{\tau}$ of the ELBO should converge to the maximum of the expected ELBO, which intuitively should be Z.

Starting from equation B.1, by linearity it suffices to substitute A by $\mathscr{A} := \mathbb{E}\left[A|Z\right]$. This matrix has a simple structure. If $J_k := \mathbf{1}_k^t \mathbf{1}_k - I_k$, then

$$\mathscr{A} = \begin{pmatrix} \gamma_{11} J_{n_1} & \gamma_{12} \mathbf{1}_{n_1 \times n_2} \\ \gamma_{12} \mathbf{1}_{n_2 \times n_1} & \gamma_{22} J_{n_2} \end{pmatrix}.$$

Notice also that $\bar{A} := J - A$ becomes

$$J - \mathscr{A} = \begin{pmatrix} (1-\gamma_{11}) J_{n_1} & (1-\gamma_{12}) \mathbf{1}_{n_1 \times n_2} \\ (1-\gamma_{12}) \mathbf{1}_{n_2 \times n_1} & (1-\gamma_{22}) J_{n_2} \end{pmatrix},$$

which is the expected adjacency matrix of complementary graphs to the graphs originating from the model. This matrix will be denoted $\bar{\mathscr{A}} := J - \mathscr{A}$. In order to organize the calculations, break the ELBO B.1 in two parts:

$$\mathcal{L}_{(k=2)} = \mathcal{L}^{\log}_{(k=2)} + \mathcal{L}^{\text{sym}}_{(k=2)}, \tag{B.10}$$

where

$$\begin{aligned}\mathcal{L}^{\text{sym}}_{(k=2)} :=& \frac{1}{2}\left(\tau^t A\tau \log\gamma_{11} + \tau^t \bar{A}\tau \log\left(1-\gamma_{11}\right)\right)\\ &+ \left(\tau^t A\left(\mathbf{1}_n - \tau\right)\log\gamma_{12} + \tau^t \bar{A}\left(\mathbf{1}_n - \tau\right)\log\left(1-\gamma_{12}\right)\right)\\ &+ \frac{1}{2}\left(\left(\mathbf{1}_n - \tau\right)^t A\left(\mathbf{1}_n - \tau\right)\log\gamma_{22} + \left(\mathbf{1}_n - \tau\right)^t \bar{A}\left(\mathbf{1}_n - \tau\right)\log\left(1-\gamma_{22}\right)\right),\end{aligned}$$

and $L^{\log}_{(k=2)}$ are the remaining non-random (in A) terms. Denote $\tau = (\tau_1, \tau_2)$, where $\tau_1 \in [0,1]^{n_1}$, $\tau_2 \in [0,1]^{n_2}$. Finally, denote $H(a,b) := a\log b + (1-a)\log 1 - b$. The symmetric term above expands to

Symmetric term.

$$\begin{aligned}\mathcal{L}^{\text{sym}}_{(K=2)} = \frac{1}{2}&\left[H(\gamma_{11},\gamma_{11})\tau_1 J_{n_1}\tau_1 + H(\gamma_{22},\gamma_{11})\tau_2 J_{n_2}\tau_2\right.\\ &\qquad\left. + 2H(\gamma_{12},\gamma_{11})\tau_1\mathbf{1}_{n_1\times n_2}\tau_2\right]\\ &+ \left[H(\gamma_{11},\gamma_{12})\tau_1 J_{n_1}\bar{\tau}_1 + H(\gamma_{12},\gamma_{12})\tau_1\mathbf{1}_{n_1\times n_2}\bar{\tau}_2\right.\\ &\qquad\left. + H(\gamma_{12},\gamma_{12})\tau_2\mathbf{1}_{n_2\times n_1}\bar{\tau}_1 + H(\gamma_{22},\gamma_{12})\right]\\ &+ \frac{1}{2}\left[H(\gamma_{11},\gamma_{22})\bar{\tau}_1 J_{n_1}\bar{\tau}_1 + H(\gamma_{22},\gamma_{22})\bar{\tau}_2 J_{n_2}\bar{\tau}_2\right.\\ &\qquad\left. + 2H(\gamma_{12},\gamma_{22})\bar{\tau}_1\mathbf{1}_{n_1\times n_2}\bar{\tau}_2\right].\end{aligned}$$

Now, it holds that $H(a,b) \leq H(a,a)$, implying (after straightforward simplification) that

$$\mathcal{L}^{\text{sym}}_{(k=2)} \leq C_{n_1}H(\gamma_{11}) + C_{n_2}H(\gamma_{22}) + n_1 n_2 H(\gamma_{12}),$$

TO DO: Put this inequality as a proposition !

using the notation from B.7. Notice the right-hand side is constant in τ. Finally, one can check that substituting $\tau^\star = (\mathbf{1}_{n_1}, 0_{n_2})$ achieves this upper bound. Thus it maximizes the symmetric part of the ELBO and corresponds to Z, the true community labels. As for the other term, notice that at $\tau^\star$ it becomes

Logarithmic term.

$$\mathcal{L}^{\log}_{(k=2)}(\tau^\star) = n_1\log\pi + n_2\log\left(1-\pi\right).$$

This is not the maximum that such term can reach, since if one takes $\tau = \pi\mathbf{1}_n$ then this nonpositive term vanishes. However, notice that in order to analyze the ELBO asymptotically, proper normalization is required. Dividing the ELBO by $C = \binom{n}{2}$, this logarithmic term vanishes (since it grows linearly on the size of the communities).

Therefore, assigning the true labels $\tau = Z$ *asymptotically* maximizes the ELBO *in expectation*.

Important conclusion.

Bibliography

[1] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *arXiv:1703.10146 [cs, math, stat]*, March 2017. arXiv: 1703.10146.

[2] Thang Nguyen Bui and Curt Jones. Finding good approximate vertex and edge partitions is NP-hard. *Information Processing Letters*, 42(3):153–159, May 1992.

[3] Shaofeng Deng, Shuyang Ling, and Thomas Strohmer. Strong Consistency, Graph Laplacians, and the Stochastic Block Model. Technical Report arXiv:2004.09780, arXiv, April 2020. arXiv:2004.09780 [cs, stat] type: article.

[4] P. Erdös and A. Rényi. On random graphs i. *Publicationes Mathematicae Debrecen*, 6:290, 1959.

[5] D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Adaptive computation and machine learning. MIT Press, 2009.

[6] Pierre Latouche, Etienne Birmelé, and Christophe Ambroise. Overlapping stochastic block models with application to the French political blogosphere, April 2011. arXiv:0910.2098 [stat].

[7] Sirio Legramanti, Tommaso Rigon, Daniele Durante, and David B. Dunson. Extended Stochastic Block Models with Application to Criminal Networks, April 2022. arXiv:2007.08569 [stat].

[8] Jure Leskovec. Stanford large network dataset collection. `https://snap.stanford.edu/data/#communities`. Accessed: 2022-08-25.

[9] H. Lutkepohl. *Handbook of Matrices*. Wiley, 1996.

[10] Mahendra Mariadassou, Stéphane Robin, and Corinne Vacher. Uncovering latent structure in valued graphs: A variational approach. *The Annals of Applied Statistics*, 4(2):715–742, June 2010. Publisher: Institute of Mathematical Statistics.

[11] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. OUP Oxford, 2009.

[12] Vincent Miele and Catherine Matias. Revealing the hidden structure of dynamic ecological networks. *Royal Society Open Science*, 4(6):170251, June 2017.

[13] Leonardo Morelli, Valentina Giansanti, and Davide Cittaro. Nested Stochastic Block Models applied to the analysis of single cell data. *BMC Bioinformatics*, 22(1):576, November 2021.

[14] Mark Newman. Network data. http://www-personal.umich.edu/~mejn/netdata/. Accessed: 2022-08-25.

[15] Tiago Peixoto. Netzschleuder. https://networks.skewed.de/. Accessed: 2022-08-25.

[16] Roldan Pozo. Complex network resources. https://math.nist.gov/~RPozo/complex_datasets.html. Accessed: 2022-08-25.

[17] Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4), August 2011. arXiv: 1007.1684.

[18] Purnamrita Sarkar, Y. X. Rachel Wang, and Soumendu S. Mukherjee. When random initializations help: a study of variational inference for community detection. *Journal of Machine Learning Research*, 22(22):1–46, 2021.

[19] A.W. van der Vaart. *Asymptotic Statistics*. Asymptotic Statistics. Cambridge University Press, 2000.

[20] Ulrike von Luxburg. A Tutorial on Spectral Clustering. *arXiv:0711.0189 [cs]*, November 2007. arXiv: 0711.0189.

[21] Yi Yu, Tengyao Wang, and Richard J. Samworth. A useful variant of the davis–kahan theorem for statisticians, 2014.