THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

Spécialité : SIGNAL IMAGE PAROLE TELECOMS

Arrêté ministériel : 25 mai 2016

Présentée par

Lorenzo DALL'AMICO

Thèse dirigée par **Romain COUILLET**,Professeur des Universités
et codirigée par **Nicolas TREMBLAY**, CNRS

préparée au sein du **Laboratoire Grenoble Images Parole Signal Automatique**
dans **l'École Doctorale Electronique, Electrotechnique, Automatique, Traitement du Signal (EEATS)**

Méthodes spectrales pour la classification sur les graphes

Spectral methods for graph clustering

Thèse soutenue publiquement le **12 octobre 2021**,
devant le jury composé de :

Monsieur Romain COUILLET
PROFESSEUR DES UNIVERSITES, UNIVERSITE GRENOBLE ALPES, Directeur de thèse
Madame Lenka ZDEBOROVA
PROFESSEUR ASSOCIE, Ecole Polytechnique Fédérale de Lausanne, Rapporteure
Monsieur Tiago DE PAULA PEIXOTO
PROFESSEUR ASSOCIE, Central European University - Vienna, Rapporteur
Monsieur Alain BARRAT
DIRECTEUR DE RECHERCHE, CNRS DELEGATION PROVENCE ET CORSE, Examinateur
Monsieur Francesco BONCHI
DOCTEUR EN SCIENCES, ISI foundation - Torino, Examinateur
Monsieur Jean-Philippe BOUCHAUD
PROFESSEUR DES UNIVERSITES, ENS PARIS, Examinateur
Monsieur Laurent MASSOULIE
DIRECTEUR DE RECHERCHE, INRIA CENTRE DE PARIS, Examinateur

SPECTRAL METHODS FOR GRAPH CLUSTERING

LORENZO DALL'AMICO
GIPSA lab - Grenoble INP

12 October 2021

ABSTRACT

Categorization, *i.e.* the ability to assign the same labels to objects sharing similar properties, is one of the main task in machine learning. In recent times, the ever increasing amount of data at our disposal gives us unprecedented possibilities to devise sophisticated and statistically significant categorization methods but it also requires a considerable effort in designing scalable and efficient algorithms, capable to properly deal with these datasets.

Spectral clustering (SC) is one of the most popular techniques to categorize the items of a dataset that can be represented as a graph. This is a class of unsupervised algorithms in which the "best" partition does not require the help of additional information to be determined and is instead obtained by exploiting the dependencies between the dataset's items. In SC algorithms, the information concerning the class structure of the input dataset is determined by the eigenvectors of a suited matrix. The intuitions and results justifying SC are at the crossroads between several fields such as statistics, random matrix theory, computer science, network science, signal processing, statistical physics and have so far mostly been treated independently.

In this manuscript, we study the challenging (but relevant) *sparse* setting, in which only few entries of the matrix representation of the input dataset are non-zero. We focus in particular on the applications of SC for community detection (in both static and dynamical graphs) and for the sparsification of kernel matrices for time-efficient clustering of high dimensional vectors. We build on the recent advances in statistical physics for SC to propose improved algorithms that provably outperform the existing methods for both synthetic as well as real clustering tasks. Moreover, we propose a simple framework that gives a unified view of some of the most influential state-of-the-art methods for SC. The existing algorithms from the literature can often be considered as corner cases of our proposed methods that constitute instead an "optimum" capable of self-adapting to the hardness of the clustering problem. We further detail extensively how to efficiently implement our proposed algorithms for practical SC tasks.

RESUMÉ

La catégorisation, c'est-à-dire la capacité à attribuer les mêmes étiquettes à des objets partageant des propriétés similaires, est l'une des principales tâches de l'apprentissage automatique. Ces dernières années, la quantité toujours croissante de données à notre disposition nous offre la possibilité sans précédent de concevoir des méthodes de catégorisation sophistiquées et statistiquement significatives, mais elle exige également un effort considérable pour concevoir des algorithmes évolutifs et efficaces, capables de traiter correctement ces ensembles de données. Le clustering spectral (SC) est l'une des techniques les plus populaires pour catégoriser les éléments d'un ensemble de données qui peut être représenté sous la forme d'un graphe. Il s'agit d'une classe d'algorithmes non supervisés pour lesquels la "meilleure" partition ne nécessite pas l'aide d'informations supplémentaires et est plutôt obtenue en exploitant les dépendances entre les éléments du jeu de données. Dans les algorithmes SC, l'information concernant la structure des données d'entrée est obtenue grâce aux vecteurs propres d'une matrice appropriée. Les intuitions et les résultats justifiant le SC sont à la croisée des chemins de plusieurs domaines tels que les statistiques, la théorie des matrices aléatoires, l'informatique, la science des réseaux, le traitement du signal, la physique statistique et ont jusqu'à présent été traités de manière indépendante.

Dans ce manuscrit, nous étudions le cadre difficile (mais pertinent) des matrices parcimonieuses, dans lesquelles seules quelques entrées de la représentation matricielle sont différentes de zéro. Nous nous concentrons en particulier sur les applications du SC pour la détection de communautés (à la fois statiques et dynamiques) et pour la sparsification des matrices à noyau pour le clustering de vecteurs en grande dimension. Nous nous appuyons pour cela sur les avancées récentes de la physique statistique pour le SC afin de proposer des algorithmes améliorés qui surpassent, preuve à l'appui, les méthodes existantes pour les tâches de classification autant sur des données synthétiques que sur des données réelles. De plus, nous proposons un cadre simple qui donne une vue unifiée de certaines des méthodes les plus influentes qui forment l'état de l'art pour le SC. Les algorithmes existants de la littérature peuvent souvent être considérés comme des cas extrêmes des méthodes que nous proposons qui constituent plutôt un "optimum" capable de s'adapter à la difficulté du problème de classification. Nous détaillons également une implémentation efficace des algorithmes que nous proposons pour les tâches pratiques de SC.

CONTENTS

ACRONYMS

SC *spectral clustering*

CD *community detection*

DCD *dynamical community detection*

RMT *random matrix theory*

ER *Erdős-Rényi*

GW *Galton-Watson*

SBM *stochastic block model*

DSBM *dynamical stochastic block model*

DCSBM *degree corrected stochastic block model*

DDCSBM *dynamical degree corrected stochastic block model*

NMF *naïve mean field*

BP *belief propagation*

RBIM *random bond Ising model*

SYMBOLS

Miscellaneous

- The cardinality of set $\mathcal{A}$ is denoted with $|\mathcal{A}|$.
- The empty set is denoted with $\emptyset$.
- $\mathbb{I}_x$ is the indicator function equal to 1 if the condition x is verified and is 0 otherwise.
- $\mathbf{1}_n$ is the all-ones vector of size n.
- When two quantities a_n, b_n admit a finite limit $\lim_{n\to\infty} \frac{a_n}{b_n} = c$, independent of n, then we write $a_n \sim b_n$, or, equivalently, $a_n = O_n(b_n)$. Conversely, if $\lim_{n\to\infty} \frac{a_n}{b_n} = 0$, then we write $a_n = o_n(b_n)$.
- δ_{ij} is the Kronecker symbol, equal to 1 is $i = j$ and 0 otherwise.

Linear algebra

- Matrices are denoted in standard font capital letters (M). The transpose of a matrix is M^T. The i-th row and column of a matrix M are denoted with $M_{i,\bullet}$ and $M_{\bullet,i}$, respectively. M_{ij} corresponds to the entry of M at row i and column j.
- Column vectors are denoted in small-case bold font ($\boldsymbol{v}$). Row vectors are denoted with $\boldsymbol{v}^T$. Vectors elements are denoted as v_i.
- Scalars are denoted in small-case standard font (a, v_i).
- I_n is the diagonal identity matrix of size n.
- $M = \text{diag}(\boldsymbol{v})$ is the diagonal degree matrix so that $M_{ij} = \delta_{ij} v_i$.
- $\text{tr}(\cdot)$ is the trace of a matrix.
- $\det(\cdot)$ is the determinant of a matrix.
- $\lambda_i(\cdot)$ indicates a generic eigenvalue of a matrix.
- For Hermitian matrices, $\lambda_i^{\uparrow}(\cdot)$ (resp. $\lambda_i^{\downarrow}(\cdot)$) is the i-th smallest (resp. largest) eigenvalue.
- For non Hermitian matrices $\lambda_{R,i}^{\uparrow}(\cdot)$ (resp. $\lambda_{R,i}^{\downarrow}(\cdot)$) is the i-th eigenvalue with smallest (res. largest) real part. Similarly, $\lambda_i^{\uparrow|\cdot|}(\cdot)$ (resp. $\lambda_i^{\downarrow|\cdot|}(\cdot)$) is the i-th smallest (resp. largest) eigenvalue in modulus.
- $\Lambda(M)$ is the set of all the eigenvalues of M.

INTRODUCTION

CATEGORIZATION AND LEARNING

Categorization, *i.e.* the ability to divide objects into groups, is deeply rooted in human intelligence and takes a prominent role in learning. One could even say that the criterion followed to categorize objects precisely provides a definition of the abstract concept related to the categories themselves [Hul20].

Whether consciously or not, human intelligence allows each one of us to perform two basic operations, sketched in the toy representation of Figure 1, to accomplish the task of categorizing objects. The first one requires to identify the informative features of the input data, *i.e.* those that are necessary and sufficient to attribute a category label to the input data. For example, considering apples and pears, discriminative features may be the taste, the texture, while the weight of the fruit would generally be non-informative. This step is called *feature extraction* and provides a representation of the object in a parametric space, so that similar objects in the real world (the apples or pears) should correspond to nearby points in the parametric space.

From humans...

The second step consists in attributing a label to the input, according to its position in the feature space. Depending on the problem considered, this can be done in two main ways. One is *classification*: the task is assigning a single given item to one of the possible available groups; the other is *clustering*: in this case, the goal is to divide into groups several input items, according to their proximity as shown in the right plot of Figure 1.

Machine learning aims at reproducing with computers the human ability of finding patterns in data [Bis06]. Machines have a decisive disadvantage with respect to humans in accomplishing complex tasks like pattern recog-

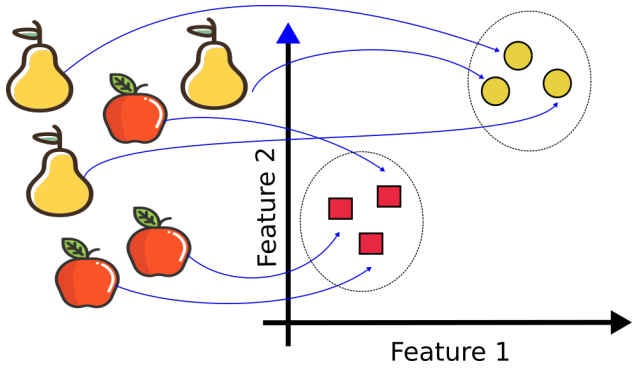


Figure 1: A toy sketch of the steps of feature extraction and clustering. The blue arrows indicate the feature extraction step, while the black circles group together nearby points in the feature space.

...to machines

nition as they are bounded to a simple representation of inputs in the form of arrays of bits and do not have a complex experience of reality that the five senses give to humans. Nevertheless the steps of *feature extraction* and successive *clustering* or *classification* are the fundamental building blocks of several machine learning algorithms [MRT18].

Supervised learning

Classification is typically achieved with *supervised learning*, which is a very intuitive (in principle) way of intending learning and is nowadays most commonly deployed. A large dataset of items equipped with their ground-truth labels is used to train an algorithm to generate a rule predicting the labels. Once the training has been completed, the rule can be used to classify newly submitted inputs. To this class of algorithms belong neural networks that were born with the aim of reproducing the learning process that happens in a human brain [MP43] and constitute nowadays the state-of-the-art in most fields in terms of performance. This approach has however its drawbacks: the learning step may be computationally expensive, while the creation of a suited training set may be time and storage consuming.

Unsupervised learning

Related to the problem of clustering instead, *unsupervised learning* techniques are generally preferred in cases where the machine is not provided with any additional information beyond the data to be clustered. The lack of "external aid" of course hampers the performances of unsupervised learning as compared to supervised learning, but the inconveniences introduced by the learning step are here bypassed. Unsupervised learning algorithms in fact profit from the ability of machines to perform hard computing tasks that is precisely where they overcome humans, being capable of processing enormous amount of data in a short time. The advantage of dealing with large datasets is indeed to exploit statistically relevant relations between its elements and to find patterns with no additional external information.

Even though supervised learning algorithms are nowadays very popular due to their ability to obtain super-human performances, serious concerns should be raised about their requirements of several labelled samples and of computational intensive training [Usa+19]. The ever increasing size of input datasets, in fact, cannot (and probably *should* not) be accompanied by an equally fast growth of computational effort. In this perspective, unsupervised techniques play a fundamental role in modern machine learning.

SPECTRAL CLUSTERING

THE BUILDING BLOCKS

Clustering has several applications in different fields and is, generally speaking, an ill-defined problem [XW05; RM05; Sch07; WK18]. One of the most popular classes of clustering algorithms, which is at the core of this manuscript, is *spectral clustering* (SC) whose building blocks are described as follows.

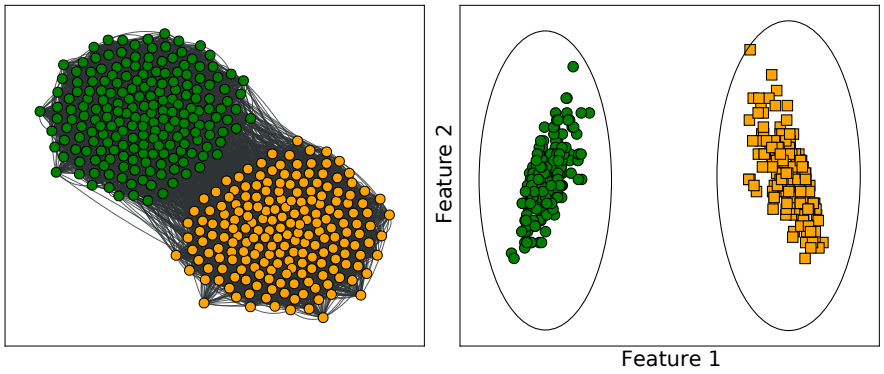


Figure 2: **Left**: a graph with two communities, highlighted with a different colour code. **Right**: a 2 dimensional node embedding of the graph on the left.

Graph representation

The first step of SC consists in providing the input dataset with a graph representation encoding the pairwise relations between its items. A more extensive introduction to graphs is dedicated in Chapter 1. For visualization purposes, a graph with two clusters is shown in the left plot[1] of Figure 2.

Feature extraction

The following step of SC is then the feature extraction, which consists in mapping each item (or *node* in graph language) into a small dimensional vector. This is also known as *node embedding* [CZC18; NS13], whose representation is given in the right plot of Figure 2 that should be compared to Figure 1. An appropriate embedding is such that "similar" nodes should end up close in the embedded space. In SC, the nodes are embedded exploiting the eigenvectors of a suited graph matrix representation. One of the main advantages of SC with respect to other embedding methods [GF18; CZC18; NS13] relies in its solid theoretical foundations and explainability.

Clustering

The final step consists in performing a low dimensional clustering on the input embedded nodes, *i.e.* in determining boundaries in the embedded space to separate the points. Several algorithms exist to accomplish this task and can be divided in two groups: those in which partitions are attributed solving an optimization problem such as *k-means* [Mac+67], *k-medoids* [KR09], *expectation maximization* [DLR77]; those in which boundaries between clusters are drawn where the density of points is minimal, as shown in Figure 3 such as *DBSCAN* [Est+96] and *OPTICS* [Ank+99].

The main steps of a typical SC algorithm are summarized in Algorithm 1. Here, it has to be noted that the dimension of the embedding is set to k, the number of clusters. This is a customary choice, but not a stringent requirement. In fact, on a more general level, Algorithm 1 is written for pedagogical reasons and not all SC algorithms follow exactly its structure.

In order to justify why Algorithm 1 can be used for clustering, we provide an example for $k = 2$ clusters, based on the work of [Fie73] that can be considered the first work on SC. The mathematical and intuitive description

1 The graph plots in this manuscript are obtained using the `graph-tool` [Pei14b] and `NetworkX` [HSS08] Python packages.

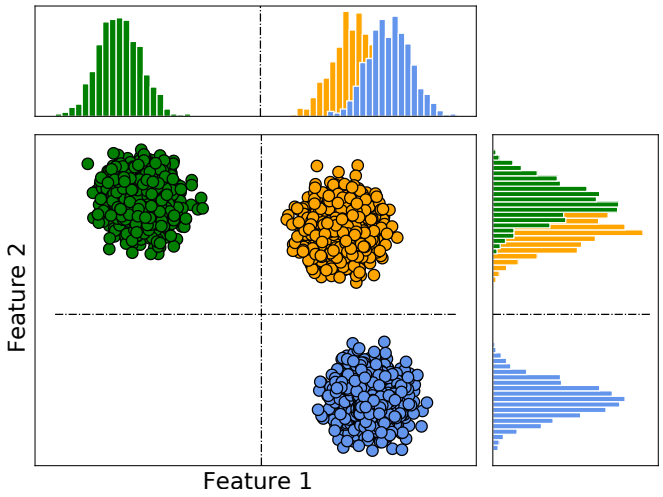
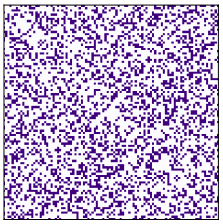


Figure 3: Three clouds of points in a 2-dimensional space. The colours indicate the ground-truth labels. The histograms represent the densities of points along the two directions. The dash-dotted black lines correspond to the minima of the histograms and allow one to cluster the input data into $k = 3$ groups using a DBSCAN-type algorithm.

is willingly left to a low level. More precise results and considerations are at the core of the main body of the manuscript.

A representation of an A: a white dot indicates $A_{ij} = 0$, while a coloured dot $A_{ij} = 1$.



Let us consider a problem in which the relations between items are encoded by binary variables $A_{ij} \in \{0,1\}$ with $A_{ij} = 0$ indicating that i and j do not interact and, on the opposite $A_{ij} = 1$ that they do. This is the simplest possible setting to describe pairwise interactions and the matrix $A \in \{0,1\}^{n\times n}$ is the *adjacency matrix* of an *unweighted graph*, that will be more formally defined in Chapter 1. Letting $D = \mathrm{diag}(A\mathbf{1}_n)$ be the diagonal *degree* matrix, where $\mathbf{1}_n$ is the all-ones vector of size n, we introduce the *graph Laplacian* matrix $L = D - A$. For any vector $\boldsymbol{f} \in \mathbb{R}^n$ we can write

$$\begin{aligned} \boldsymbol{f}^T L \boldsymbol{f} &= \sum_{i,j=1}^{n} f_i \left(d_i \delta_{ij} - A_{ij}\right) f_j = \sum_{i=1}^{n} f_i^2 d_i - \sum_{i,j=1}^{n} A_{ij} f_i f_j \\ &= \sum_{i,j=1}^{n} A_{ij} \left(f_i^2 - f_i f_j\right) = \frac{1}{2} \sum_{i,j=1}^{n} A_{ij} \left(f_i - f_j\right)^2, \end{aligned} \tag{1}$$

where δ_{ij} denotes the Kronecker delta. Any pair of interacting nodes ($A_{ij} = 1$) contributes to this sum with a weight which increases with the distance between f_i and f_j. Now, supposing $\boldsymbol{f}$ is a node embedding vector, $(f_i - f_j)^2$ should be small if i and j are "similar" and large otherwise. Consequently $\boldsymbol{f}^T L \boldsymbol{f}$ can be seen as an embedding cost function. In fact, assuming that nodes are more likely to interact if they are similar, to large values of $\boldsymbol{f}^T L \boldsymbol{f}$ correspond vectors in which interacting nodes have very different embeddings. Minimizing $\boldsymbol{f}^T L \boldsymbol{f}$ under the constraint $\boldsymbol{f} \perp \mathbf{1}_n$ (which is a trivial embedding) allows one to map "similar" nodes to "nearby" points. Imposing the normalization $\|\boldsymbol{f}\|^2 = 1$, we define the embedding vector $\boldsymbol{f}^*$ as

Algorithm 1 : Spectral clustering

Input : Dataset with n items, k number of clusters
Output : $\ell \in \{1, \dots, k\}^n$ label assignment

```
begin
    Define suited matrix representation of the dataset M ∈ ℝ^{n×n};
    Stack the k largest (or smallest) eigenvalues of M in the columns
      of X ∈ ℝ^{n×k} (Feature extraction);
    Estimate community labels ℓ with a small dimensional clustering
      algorithm performed on the rows of X (Clustering) ;
    return ℓ
end
```

$$f^* = \arg \min_{f \in \mathbb{R}^n, f \perp \mathbf{1}_n} f^T L f.$$

Since $\mathbf{1}_n$ is the eigenvector of L associated to its smallest eigenvalue (this is easily seen from Equation (1)), f^* is the eigenvector associated to the second smallest eigenvalue of L, known as *Fiedler eigenvector*, and it justifies SC in the form of Algorithm 1 for the choice $M = L$. Following similar arguments, some classical SC algorithms are based on the adoption of the *normalized Laplacian* matrices $L^{\text{sym}} = D^{-1/2} A D^{-1/2}$, $L^{\text{rw}} = D^{-1} A$ and the adjacency matrix A [VL07; LR+15], with possible extensions also to the weighted case in which $A \in \{0,1\}^{n \times n}$ is replaced by $W \in \mathbb{R}^{n \times n}$.

A BIRD'S EYE VIEW ON SC

The field of SC is very interdisciplinary and, in order to describe at best where the work presented in this manuscript locates in the existing literature, a wide perspective on the problem must be kept.

The computational complexity of SC

First of all, as we mentioned earlier, one of the main interests of SC lies in its solid theoretical foundation. Powerful tools based on *random matrix theory* (RMT) allow one to provide precise results on the performances and applicability of Algorithm 1. Typically, however, two important hypotheses have to be formulated: the size n of the matrix M must go to infinity and the number of non-zero entries (say m) must go to infinity faster than n [CBG+16; AC17, for instance]. While the former simply consists in considering the very interesting setting of big datasets, from a practical perspective, the latter hypothesis raises a problem from the point of view of the scalability of Algorithm 1. In fact, the numerical cost of computing the k leading eigenvectors of a matrix with m non-zero entries grows as $O(mk^2)$ [Saa92]. Let us consider the case $m = n^2$, *i.e.* all the entries of the matrix M are non-zero, for a small value of k such as $k = 2$. The complexity is in this case $O(n^2 k^2)$ which makes the cost of Algorithm 1 prohibitive for a standard laptop when $n > 10^5$ approximately.

The problem of the computational complexity of SC is well known in the literature and led to several sampling strategies that aim at improving the efficiency of Algorithm 1 [TL20]. The simplest procedure simply consists in making the matrix M sparser, setting some of its entries to zero. This improves significantly the efficiency of Algorithm 1. Consider in fact a sampling strategy that keeps on average 10 non-zero entries in each row of M: the computational limit of the laptop considered earlier is now obtained for $n = 10^9$ which allows one to deal with datasets of considerable sizes.

SC and sparsity

This sampling procedure, however, comes at a cost. Besides the fact that by "throwing away" some measurements (hence information), the output of Algorithm 1 may naturally be less accurate, sparsity has long known to be the "Achilles' hill" of SC. In fact, for great levels of sparsification, while there exist algorithms different from SC that can find reasonable clusters, Algorithm 1 typically performs very poorly for the standard choices of the matrix M, such as the *graph Laplacian* $L = D - A$ introduced earlier.

The problem of sparsity is not only related to sampling procedures. A natural application of SC is to perform node embeddings on social networks in order to divide the nodes into *communities*. Real social networks are often sparse by construction [Bar13] (each node interacts typically with a very small fraction of the total) and, as a consequence, Algorithm 1 may perform poorly in practical *community detection* (CD) tasks. All in all, these are severe limitations of SC that make it unsuited for a large set of inputs (such as the aforementioned social networks) and that allow for good performances only at the price of a great computational cost.

A breakthrough work on SC in sparse graphs was however provided in [Krz+13], "redeeming" the use of SC techniques of very sparse graphs, specifically in the context of CD. The work of [Krz+13], together with [Dec+11; SKZ14], is based on very deep but non-rigorous intuitions and methods borrowed from statistical physics that renewed the interest of the scientific community in the topic and led, in the following years, to formalize most of their results in a mathematically rigorous way. As far as SC is concerned (and in particular CD), the main contribution of [Krz+13; SKZ14] consists in proposing "new" matrices[2] M for Algorithm 1 to replace the "classical" choices when performing clustering in sparse graphs.

Statistical physics gave powerful insights on SC in sparse graphs

The Bethe-Hessian matrix plays a fundamental role in this manuscript, being the choice for M in all our proposed algorithms. In the unweighted case, its definition reads, for a scalar $r \geq 1$

The Bethe-Hessian matrix

$$H_r = (r^2 - 1)I_n + D - rA,$$

where I_n is the identity matrix of size n. We now attempt to describe in simple terms the fundamental role played by the parameter r, adopting also in this case a willingly hand-waving tone and leaving rigorous details to the following chapters.

2 The non-backtracking matrix of [Krz+13] and the Bethe-Hessian of [SKZ14] were known already, but were not used for CD.

The "statistical-physics" approach is deeply related to Bayesian inference in which, given the observation of A, each label configuration $\boldsymbol{\ell} = \{1, \dots, k\}^n$ has a probability to be the "good" label assignment given by

$$\mathbb{P}(\boldsymbol{\ell}|A) = \frac{1}{Z} \exp\left\{-\text{ath}\left(\frac{1}{r}\right) \mathcal{H}(\boldsymbol{\ell})\right\},$$

where $\mathcal{H}(\boldsymbol{\ell})$ is a cost function of the label assignment and Z a normalization constant. For $k = 2$ classes,[3] the eigenvector associated to the second smallest eigenvalue of H_r can be used to obtain an approximation of the vector of marginal node probabilities $\{P_i(\ell_i)\}_{i=1,\dots,n}$. It is now fundamental to understand the role played by r. For $r \to 1$, only the configuration minimizing the cost function $\mathcal{H}(\boldsymbol{\ell})$ matters and the problem of computing the node marginals simply reduces to finding the optimum of $\mathcal{H}(\boldsymbol{\ell})$. For larger values of r, instead, each configuration $\boldsymbol{\ell}$ contributes to the function $\mathbb{P}(\boldsymbol{\ell}|A)$ and prevents the marginals from being potentially determined by meaningless local minima of $\mathcal{H}(\boldsymbol{\ell})$ introduced by noise: the higher the noise, the larger r should be. In simple words, therefore, tuning r in H_r allows one to avoid over-fitting.

This concludes the brief introduction to the many facets of SC considered in this manuscript which is a metaphorical dance between powerful physics intuitions, rigorous mathematical results and practical questions of applicability of our proposed algorithms to real-world problems.

OUTLINE AND CONTRIBUTIONS

In order to provide the reader with the necessary tools needed to best understand our contributions, Part I introduces the technical results exploited in the remainder. In particular, Chapter 1 provides an introduction to graphs, together with their definitions and properties. Then, Chapter 2 gives the fundamental tools of statistical physics on which our main findings rely. Finally, Chapter 3 introduces some important mathematical results on the spectral properties of two matrices of fundamental importance in this manuscript: the non-backtracking and the aforementioned Bethe-Hessian matrices.

Part II is then dedicated to the presentation of our original contributions to SC for CD and is opened by Chapter 4 that provides an extensive introduction to the problem.

Here we take the "torch" of our predecessors and address the problem of how overly simplistic assumptions may result in poor algorithmic performances. Specifically, in [SKZ14], the authors proposed a very powerful spectral algorithm for CD in very sparse graphs, based on the Bethe-Hessian matrix H_r. Their theoretical results are formulated under the assumption

3 We consider $k = 2$ for simplicity, but we will show in the remainder that H_r can be used for SC in the presence of $k \geq 2$ classes.

that the matrix A is generated from the *stochastic block model* (SBM), according to which each node of the graph has approximately the same number of connections. Real social networks (in which CD is of utmost importance), however, are known to be typically heterogeneous, in the sense that the number of connections each node has varies a lot across the network. As a consequence of this overly simplifying assumption, the algorithm of [SKZ14] (that prescribes a precise choice of r) often performs poorly compared to classical SC algorithms (see Table 7.1).

Our earliest works are presented in Chapter 5 and explain how we improved over [SKZ14], considering a more realistic generative model, called *degree corrected stochastic block model* (DCSBM) which indeed allows one to keep the graph heterogeneity into account. The first of these works is [DC19]

> **LD**, Romain Couillet: *Community detection in sparse realistic graphs: Improving the Bethe-Hessian* in ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (2019)

in which we proposed (for $k = 2$ classes of equal size) an alternative choice of r from the one of [SKZ14], able to provide high performance clustering on heterogeneous graphs. The proposed parametrization r, however, requires the knowledge of some of the parameters of the generative model of the graph that cannot be given for granted.

With our next contribution [DCT19]

> **LD**, Romain Couillet, Nicolas Tremblay: *Revisiting the Bethe-Hessian: Improved Community Detection in Sparse Heterogeneous Graphs* in NeurIPS Advances in Neural Information Processing Systems 32 (2019)

we answered some of the main open questions of [DC19] . First, we extended our results to the setting of $k \geq 2$ classes of arbitrary size, showing that a sequence of properly chosen parameters r must be selected to get an efficient SC algorithm on heterogeneous graphs. Secondly, we showed that these parameters r can be estimated in an unsupervised way. We did so unveiling an unforeseen property of the eigenvalues of the non-backtracking matrix B used in [Krz+13] and exploiting the known relation between B and H_r determined by the Ihara-Bass formula [Ter10]. Finally, backed by several arguments and numerical simulations, we claimed our proposed parametrization to be *optimal*, in the sense that no other parametrization is expected to perform better on DCSBM-generated graphs.

Having a practical way to estimate the optimal values of r in an unsupervised way further allowed us to devise and test a practical algorithm for CD. The results confirm that the improved parametrization achieves a systematic higher performance with respect to [SKZ14] on all tested datasets.

In the SC literature, however, not only the contributions of the physics community led to very efficient algorithms on sparse graphs. In fact, a notable and independent line of work, with an approach rooted in statistics,

proposed the use of new *regularized* matrices for SC. In this introduction we only mention one these matrices, $L_\tau^{\text{sym}} = D_\tau^{-1/2} A D_\tau^{-1/2}$, proposed in [QR13] with $D_\tau = D + \tau I_n$ for τ equal to the graph average degree. This is *de facto* the state-of-the-art of SC for CD on real graphs. In Chapter 6 we present our work related to L_τ^{sym}, started in [DCT20c]

> **LD**, Romain Couillet, Nicolas Tremblay: *Optimal Laplacian regularization for sparse spectral community detection* in ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (2020)

in which we showed that there exists a clear relation between the Bethe-Hessian matrix H_r and the *regularized Laplacian* L_τ^{sym}. Furthermore, based on our results in [DC19; DCT19] we showed that the choice of τ of [QR13] is suboptimal. The optimal Laplacian regularization is obtained with a set of τ closely related to the r's to be adopted for H_r and that can be estimated from the graph in an unsupervised fashion. Matter of factly, some of the results presented in Chapters 5, 6 are taken from [DCT20a]

> **LD**, Romain Couillet, Nicolas Tremblay: *A unified framework for spectral clustering in sparse graphs* accepted to Journal of machine learning research

in which we summarized our results in a *unified framework* making formal statements of our conjectures and providing rigorous proofs to part of them. A further big contribution of [DCT20a] is at the centre of Chapter 7 and concerns the applicability of our proposed algorithms on practical real graphs as well as their efficient implementation. Here we propose Algorithm 7.1 for CD and conduct a systematic study on real datasets showing the efficiency of our proposed method for practical tasks. On top of this we also released an efficient implementation in `Julia` language of Algorithm 7.1 for CD, called `CoDeBetHe.jl` (**Co**mmunity **De**tection with the **Bet**he **He**ssian).

Chapter 8 finally closes part II starting from the results posed in [DCT20a] to provide a unified view of several SC methods that have so far been treated independently. In this chapter, in fact, we clearly show that our proposed algorithm is *adaptive* to the hardness of the clustering task and that common state-of-the-art methods in both the sparse and dense regime can be understood as limiting (suboptimal) cases of Algorithm 7.1.

These results pose solid methodological foundations that led us to work on the extensions of our works to more general settings than CD, treated in Part III. The first problem we considered concerns *dynamical community detection* (DCD), which enriches the description by keeping the dynamical nature of real-world graphs into account. In [DCT20b]

> **LD**, Romain Couillet, Nicolas Tremblay: *Community detection in sparse time-evolving graphs with a dynamical Bethe-Hessian* in NeurIPS Advances in Neural Information Processing Systems 33 (2020)

we considered a sequence of graph snapshots at different times with the assumption that community labels may change across time, while having a

positive correlation. We developed a novel SC algorithm based on a dynamical Bethe-Hessian matrix which is capable of exploiting the information coming from these correlations and that largely outperforms the state-of-the-art competitors. The results of [DCT20b] are treated in Chapter 9 and the corresponding algorithm is also part of the `CoDeBetHe.jl` package.

Finally, the work of [DCT21]

> **LD**, Romain Couillet, Nicolas Tremblay: *Nishimori meets Bethe: a spectral method for node classification in sparse weighted graphs* in JOURNAL OF STATISTICAL MECHANICS: THEORY AND EXPERIMENT

is presented in Chapter 10 and treats the extension of our works to the setting of matrix sparsification to improve the efficiency of SC, while keeping high performances. Testing our proposed algorithm on the prototypical problem in machine learning of clustering a set of images of dogs and cats, we showed that almost perfect accuracy can be achieved even for dramatic levels of sparsification, making our algorithm suited for clustering very large datasets. The main result of [DCT21] , however, is deeply related to statistical physics, describing an explicit relation between the Bethe approximation (from which the Bethe-Hessian matrix is derived) and the *Nishimori temperature* [Nis81], of fundamental importance in the physics of *spin-glasses* and in Bayes-optimal inference [Iba99].

Together, these works offer a new vision and robust adaptive algorithms, linking statistical physics and more conventional SC methods and able to handle unsupervised clustering in sparse, heterogeneous and dynamical graphs.

We now proceed with the technical chapters of the manuscript that introduce the necessary tools to fully describe our contributions.

TECHNICAL TOOLS

This part is devoted to the introduction of the technical tools needed to describe the original work presented in this manuscript. Notably, it is composed of three chapters. The first one is dedicated to the properties of graphs, the second one to statistical physics methods adapted to sparse graphs and, finally, the last one to the study of the spectra of the non-backtracking and Bethe-Hessian matrices.

1

AN INTRODUCTION TO NETWORKS AND GRAPHS

Abstract

This chapter introduces networks *and some of their recurrent properties observed in real-world settings.* Graphs *are then defined as the mathematical representation of networks and three probabilistic graph models (namely, the* Erdős-Rényi *(ER), the* stochastic block model *(SBM) and the* degree corrected stochastic block model *(DCSBM) models) are introduced. These models are useful to generate random graphs reproducing some of the properties typically observed in real-world networks and will be key to define our main results in the remainder.*

This chapter serves as an introduction to networks, which are a key protagonist of this manuscript. Networks can be defined as a set of items, called *nodes* together with a set of connections between the nodes, called *edges* [New03; BH11]: an example of a small network is represented in the left plot of Figure 1.1. This informal definition evidences that networks are a versatile tool to characterize systems of several interacting items and, as a consequence, they have been adopted to model a wide range of systems, such as social networks [Sco88; WF+94], technological networks [Ama+00], spatial networks [Bar11], biological networks [AA03] and many others.

Our main interest in networks revolves around the problem of creating *node embeddings*, *i.e.* to provide a small dimensional representation to the elements of a network. The goal of this operation is to "translate" a mathemat-

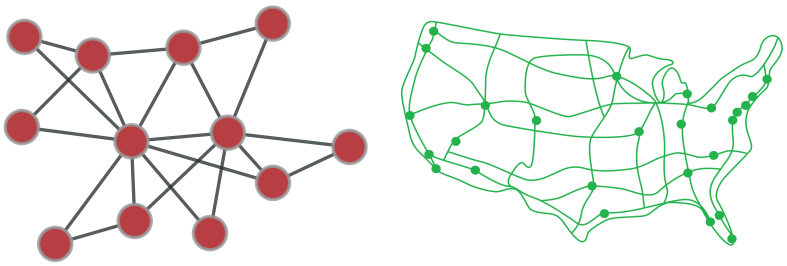


Figure 1.1: **Left**: an example of a graph with $n = 12$ nodes. Each red dot indicates a node, while each black line represents an edge. **Right**: major highways in the USA, taken from [Baro3].

ically complicated object such as a network, into a set of small dimensional vectors, preserving at least some of the network's properties.

For this reason, to fully address our main contributions and to settle the "vocabulary" of the following chapters, we must first proceed with a description of networks. We do so following an ideal path leading from observation to theory. In particular, Section 1.1 can be seen as the *observation* step in which *networks* are introduced as *physical* objects and their main characteristics as well as their possible representations are discussed. Subsequently, Section 1.2 takes the role of a *formalization* step, defining *graphs* as the *mathematical* representation of networks and laying out the main mathematical definitions. The third step is the *model creation*, necessary for a thorough analytical study of graphs. This is the content of the concluding Section 1.3 that introduces simple models – of central importance in the remainder – to generate random graphs, together with their main properties and relate these random models to the real networks discussed in Section 1.1.

1.1 NETWORKS

1.1.1 DEFINITION AND REPRESENTATIONS

To best understand networks, we now proceed to give some practical examples of real-world systems that can be represented as networks, together with their main representations.

Undirected and unweighted networks

We start considering the simplest family of networks which are *undirected and unweighted*. The toy representation in the left plot of Figure 1.1 falls precisely in this class in which the relation between two nodes is solely encoded by the existence of an edge connecting them. The term *unweighted* stands indeed for the fact that each edge does not bring any additional information beyond its existence. The term *undirected*, instead, signifies that a symmetry is assumed in the interaction of the two nodes, hence that if node i interacts with node j, than node j interacts with node i.

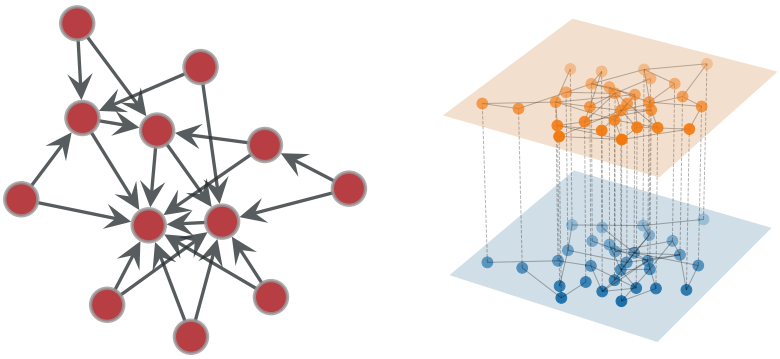


Figure 1.2: **Left**: an example of a directed graph with $n = 12$ nodes. Each red dot indicates a node, while each black arrow represents an edge together with the direction in which the edge is pointing. **Right**: a toy example of a multiplexed network with two layers indicated by different colours.

The representation of a complex system of, for example, people, with an unweighted and undirected network might come at the cost of a significant simplification and the question of what kind of interaction should be encoded by edges has no simple answer [Zwe14]. Nevertheless, a lot of systems can be modelled with unweighted and undirected networks. Some examples are given by *social networks*, *the world wide web* and *biological networks* [Sco88; WF+94; New03; AB02]. The remainder of the manuscript focuses mainly on this class of networks in which information is only encoded in the edge configuration, hence in how nodes are connected among themselves.

A richer representation that goes beyond the binary information of whether or not two nodes interact is obtained with *weighted* networks [Bar+04]. In this case, each edge carries additional information on the type of relation between the nodes, in terms of, *e.g.*, intensity of affinity. In some networks weights are easily defined, for example when considering spatial networks [Bar11], like the one shown in the right plot of Figure 1.1. There, each node represents a city, each road an edge and a weight can be attributed to each edge as function, for example, of the length of the road itself. Another example is the network `Slashdot Zoo` [KLB09] in which users express a positive or negative endorsement with respect to other users leading to a *signed network* with positive and negative weights.

Weighted networks

A limitation of both classes of networks just introduced is that they assume commutativity in the relationships between items, *i.e.* that the interaction between a and b is the same as the one between b and a. If this hypothesis is not verified, then it is more appropriate to recur to *directed* networks in which each edge encodes an orientation of the relationship between nodes, as shown in the left plot of Figure 1.2. For reference, one could consider a network in which an edge (ab) goes from node a to node b if node a sent an email to node b [NFB02]. It appears natural that the edge (ba) has a differ-

Directed networks

ent meaning from (ab), justifying the need of a directed network to describe relationships in this case. Other examples of directed networks are for example *citation networks* [TS92] for which the edge (ab) is a citation of author b from author a, or the links on the webpages on the internet [Bar03]. A natural consequence of adding directionality to the edges is to be capable of grasping the *causality* of the relations and to describe how information flows on the network.

Multiplexed and temporal networks

The last network representation we consider takes into account that in the wide range of systems that can be modelled by networks, it may occur that the relationship between the nodes of a network is not measurable only in terms of a single parameter. Suppose, for instance, one wants to determine the affinity between people. Many factors should be kept into account like, for example, age, geographical provenance, spoken language, social extraction, political orientations and so on. The affinity between items is not necessarily determined by a simple combination of the affinities at different levels, but should rather take a vector form, allowing for the possibility to have a large affinity according to some parameters but not to others. The use of *multiplexed* networks [Kiv+14] (or multilayer networks) is most suited to this scenario: in this case, multiple representations of the same network are given, encoding information at different levels as in the example shown in the right plot of Figure 1.2. For reference, a practical example of a real-world multiplexed network is the one showing parental and economical relations between the families of Florence during Renaissance [BP86] or by transportation networks in which each layer corresponds to a different line, such as bus, tram, underground lines [Kiv+14].

As a particular class of multiplexed networks one can consider *temporal* networks [HS12] in which the different representations of the network simply indicate its evolution across time. Most real-world networks are indeed dynamical and some examples are found in *human proximity networks*, *biological networks*, *communication networks* among the others. The study of temporal networks will be at the centre of Chapter 9.

After this short introduction on the different types of networks and some examples of systems that can be modelled by networks, we now proceed with the characterization of some properties that occur in several real-world networks with a specific focus to the unweighted and undirected case.

1.1.2 PROPERTIES OF REAL NETWORKS

As we detailed in the previous paragraph, networks can represent fundamentally different systems hence the emergence of some universal behavior is not an obvious fact. It appears instead that many real-world networks share some peculiarities, regardless of the underlying system. We now proceed to present some of these properties, while keeping in mind that they should not be intended as *laws*, but rather, very recurrent observations.

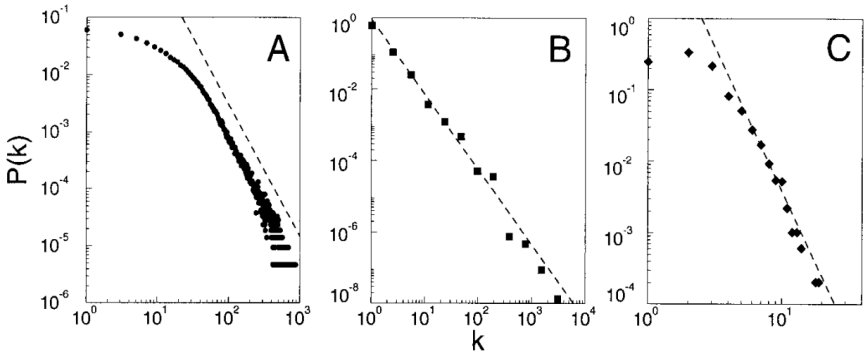


Figure 1.3: Empirical degree distribution $P(k)$. **A**: actor collaboration graph. **B**: The World Wide Web. **C**: Power grid network. Picture taken from [BA99].

The first aspect that typically characterizes real-world networks is the so-called *small world effect* [WS98]. This property is verified when, considering a network composed by n items, the average distance (*i.e.* the number of interposing nodes) between two arbitrary nodes grows as $\log(n)$ [Bar03] (or even slower). This effect is observed in sociology with the fact that humans are on average "six handshakes away" from one-another [Mil67; Gua90]. The small world effect is observed way beyond social networks, for example in the World Wide Web [Bar03]. As a consequence of this effect, information flows rapidly on networks and an emerging behavior can be observed.

Small world effect

Another peculiar aspect characterizing real-world networks is the *degree distribution*. It has been observed in a vast number of networks that the number of connections each node has (we call this number *the degree*) is far from a homogeneous distribution [BA99] (see Figure 1.3). Typically, this distribution is, instead, a power law. In this case we talk about *scale free networks*. When dealing with a broad degree distribution, most nodes have very few connections while very few are highly connected. These nodes play a prominent role due to their high connectivity and are called *hubs*.

The degree distribution

Remaining on the properties of the degree distribution, in real-world networks the average degree is typically very small compared to its size. This phenomenon goes under the name of *sparsity*. In others words, out of all the possible connections that can take place, only a very small fraction are typically present [Bar13]. Consequently, for a large network, the average degree can be considered as roughly independent of the size of the network itself.

Sparsity

The final property we wish to introduce concerns the *clustering effect*. This consists in the formation of small groups of highly connected items as a consequence of the fact that, in social networks, connections are formed on a homophilic principle, *i.e.* they are created between items sharing similar properties [LM+54; MSLC01]. Due to the weak transitivity of homophily[1] [BH11], highly connected *communities* emerge in networks.

The clustering effect

1 If A as a homophilic relation with B and B with C, then it is likely that A has a homophilic relation with C. The expression *weak transitivity* is used to denote that the relationship is only likely to be homophilic, while transitivity would lead to an implication.

These four properties represent some fundamental characteristics of real world networks, considered in the remainder. The clustering effect and the consequent emergence of a community structure is, in particular, what primarily justifies the task of node clustering. Sparsity and heterogeneity of the degree distribution are instead challenging features to deal with when performing node clustering.

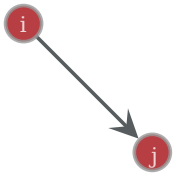
With this introduction on the main properties of real-world networks at hand, we proceed to define *graphs* as the mathematical representation of networks and laying out some formal definitions.

1.2 FROM NETWORKS TO GRAPHS

1.2.1 DEFINITIONS

To formally characterize graphs as the mathematical representation of networks, only single-layered graphs are considered: multiplexed and temporal graphs can be seen as an extension of single-layer graphs and are not formally introduced here. Further discussions on temporal graphs will follow in Chapter 9. Let us first formally introduce the definition of a graph.

A directed edge (ij)



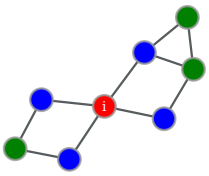
Definition 1.1 (Graph). *A graph $\mathcal{G}(\mathcal{V}, \mathcal{E}_\mathrm{d})$ is a tuple $(\mathcal{V}, \mathcal{E}_\mathrm{d})$ with $\mathcal{V}$ the set of n nodes (or vertices) and $\mathcal{E}_\mathrm{d}$ the set of edges connecting the nodes. For all $e \in \mathcal{E}_\mathrm{d}$ a weight $\omega_e \in \mathbb{R}$ can be associated. The set of weights is $\mathcal{W} = \{\omega_e\}_{e \in \mathcal{E}_\mathrm{d}}$.*

The set $\mathcal{E}_\mathrm{d}$ denotes the ensemble of all the *directed* edges of $\mathcal{G}(\mathcal{V}, \mathcal{E}_\mathrm{d})$. Each element $e \in \mathcal{E}_\mathrm{d}$ can be equivalently written as (ij), representing a directed edge from i to j. The inverse of (ij) is (ji) and is denoted by e^{-1}. For an undirected graph, if $e \in \mathcal{E}_\mathrm{d}$ then $e^{-1} \in \mathcal{E}_\mathrm{d}$. The set of *undirected* edges is denoted with $\mathcal{E}$ (with $|\mathcal{E}_\mathrm{d}| = 2|\mathcal{E}|$) and an undirected graph with $\mathcal{G}(\mathcal{V}, \mathcal{E})$. In the sequel we consider only graphs in which no self-edge exists, hence verifying for all $i \in \mathcal{V}$, $(ii) \notin \mathcal{E}$.

If $\mathcal{G}(\mathcal{V}, \mathcal{E}_\mathrm{d})$ is an unweighted graph, then $\omega_e = 1$ for all e. Note that we allow $\omega_e \in \mathbb{R}$: this is not a customary choice as often the weight is required to be strictly positive. We consider this more general case however, because it is particularly handy for our introduction to statistical physics in Chapter 2, as well as our work concerning sparsification of kernel matrices for spectral clustering presented in Chapter 10. In any case, the following definitions do not involve the weights but only the structure of $\mathcal{G}(\mathcal{V}, \mathcal{E}_\mathrm{d})$.

In red the node i, in blue its neighbours, in green the remaining nodes

We now introduce the concept of *adjacency* on a graph.



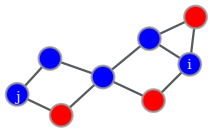
Definition 1.2 (Adjacent nodes and edges). *Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}_\mathrm{d})$, $j \in \mathcal{V}$ is said to be an* adjacent *node of $i \in \mathcal{V}$ if $(ij) \in \mathcal{E}_\mathrm{d}$. Two directed edges $(ij), (kl)$ are said to be* adjacent *if $j = k$.*

The set of all adjacent nodes of $i \in \mathcal{V}$ forms the *neighbourhood* of i.

Definition 1.3 (Neighbourhood and degree). *The* incoming neighbourhood *of a node i is defined as $\partial_{\text{in}} i = \{j \in \mathcal{V} \ : \ (ji) \in \mathcal{E}_{\text{d}}\}$, while the* outgoing neighbourhood *is $\partial_{\text{out}} i = \{j \in \mathcal{V} \ : \ (ij) \in \mathcal{E}_{\text{d}}\}$. The size of these two neighbourhoods are called* incoming degree, *$d_i^{\text{in}} = |\partial_{\text{in}} i|$, and* outgoing degree, *$d_i^{\text{out}} = |\partial_{\text{out}} i|$. For undirected graphs $\partial_{\text{in}} i = \partial_{\text{out}} i \equiv \partial i$ and the degree is defined as $d_i = |\partial i|$.*

If a node has no neighbours, we say it is *isolated*. On an undirected graph this corresponds to the condition $d_i = 0$. Neighbours are nodes that can be reached in one step. Exploiting this relation, we now introduce *paths*.

In blue, a simple path connecting i and j



Definition 1.4 (Paths). *A* path *between i and j is an ordered sequence of adjacent edges starting from i and ending in j, $\mathcal{P}_{ij} = \{(ik), (k\cdot), \ldots, (\cdot l), (lj)\}$. The length of the path is $|\mathcal{P}_{ij}|$. A path is* simple *if no edge appears twice.*

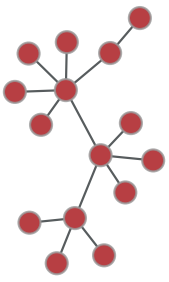
Note that on undirected graphs, all paths can be followed in reversed order and, if a path exists between two nodes, then they are said to be *connected*.

Definition 1.5 (Connected nodes). *On an undirected graph, if there exists a path $\mathcal{P}_{ij}$ from i to j (or equivalently from j to i), then i and j are said to be* connected. *The connected components of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ are the disjoint sets of all nodes that are pairwise connected. If there is a unique connected component equal to $\mathcal{V}$, then $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is said to be a* connected graph.

This definition exploits the fact that connectivity is commutative on undirected graphs. On the opposite, this is not the case for directed graphs in which there may exist a path from i to j, but not from j to i. We talk in this case about *weak connectivity* if there is a path $\mathcal{P}_{ij}$ *or* $\mathcal{P}_{ji}$ and of *strong connectivity* if both $\mathcal{P}_{ij}$ *and* $\mathcal{P}_{ji}$ exist.

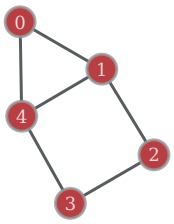
The last definition provided is that of *trees* that play a crucial role in the following chapters. In order to define trees, *cycles* need to be introduced: a simple path that starts and ends from the same node, is called a *cycle*.

A tree



Definition 1.6 (Tree). *An undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is said to be a* tree *if it is connected and it does not contain any cycle.*

The particular interest in trees in this manuscript is a consequence of their structural relation with *conditional independence* [WJo8]. Specifically, suppose that a random variable x_i is associated to each node $i \in \mathcal{V}$ and that the dependences between these variables are determined by the edge structure. As a consequence of the fact that a tree is a connected graph, for any pair of nodes, $i, j \in \mathcal{V}$, there exists a path connecting them and x_i and x_j are not independent. On a tree, however, this path is unique and if any node of the path connecting i to j (supposing they are not neighbours) is removed, then i and j are no longer connected and x_i and x_j become independent. This structural property of tree allows one to say that for any $k \in \mathcal{P}_{ij}$, the variable x_i depends on x_j only through x_k, hence x_i and x_j are independent, *conditionally* to the value of x_k. This fact makes it particularly easy to deal with dependences between random variables on tree nodes and will be further exploited in the remainder.



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

Figure 1.4: **Left**: an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $n = 5$ nodes. **Right**: the adjacency matrix A corresponding to $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

Now that we have introduced the main definitions concerning graphs, we turn our attention to their matrix representation.

1.2.2 MATRIX REPRESENTATION OF GRAPHS

Let us introduce two fundamental matrices that can be used to represent unweighted graphs. These are, namely, the *adjacency* and the *non-backtracking* matrices and will be of central importance in the remainder.

The adjacency matrix

The adjacency matrix (denoted as A) is the most straightforward representation of a graph. In words, for each pair of nodes i, j, the entry A_{ij} indicates whether the two nodes are adjacent or not. Definition 1.7 formally introduces the adjacency matrix.

Definition 1.7 (Adjacency matrix)**.** *The adjacency matrix of a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}_{\mathrm{d}})$ with n nodes is denoted with $A \in \{0,1\}^{n \times n}$ with entries*

$$\forall\, i, j \in \mathcal{V} \qquad A_{ij} = \mathbb{I}_{(ij) \in \mathcal{E}_{\mathrm{d}}},$$

where the notation $\mathbb{I}_x$ denotes the indicator function, equal to one if condition x is verified and zero otherwise. For an undirected graph, A is symmetric.

A visualization of A and the corresponding graph is given in Figure 1.4. For a given graph, the matrix A is uniquely defined and vice-versa. For this reason we say A is a representation of $\mathcal{G}(\mathcal{V}, \mathcal{E}_{\mathrm{d}})$.

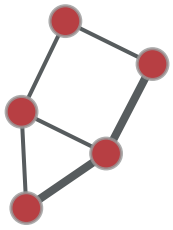
The adjacency matrix is not the only possible matrix representation of a graph: an alternative, less conventional, representation (of fundamental importance in the following) is provided by the *non-backtracking* matrix that is defined on the set of edges (instead of nodes) of the graph.

The non-backtracking matrix

The *non-backtracking* matrix, denoted with B and also known as Hashimoto edge operator [Has89], can be seen as the *directed adjacency matrix* of a

graph $\mathcal{G}'(\mathcal{V}', \mathcal{E}'_{\mathrm{d}})$ constructed as follows: each *directed* edge in $\mathcal{E}_{\mathrm{d}}$ is mapped to a node in $\mathcal{V}'$ and an edge is drawn in $\mathcal{E}'_{\mathrm{d}}$ between the nodes corresponding to $e_1, e_2 \in \mathcal{E}_{\mathrm{d}}$ if e_1 and e_2 are adjacent and $e_1 \neq e_2^{-1}$, *i.e* if e_2 is not the same edge as e_1 with reversed orientation. The reason of the name *non-backtracking* comes from the fact that edges in $\mathcal{G}'(\mathcal{V}', \mathcal{E}'_{\mathrm{d}})$ correspond to paths of length 2 in $\mathcal{G}(\mathcal{V}, \mathcal{E}_{\mathrm{d}})$ without a *backtrack*, *i.e.* paths that cross the same edge in the two directions. The matrix B is defined as follows:

Highlight on two adjacent edges that do not form a backtrack



Definition 1.8 (Non-backtracking matrix)**.** *For a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}_{\mathrm{d}})$, the non-backtracking matrix $B \in \{0,1\}^{|\mathcal{E}_{\mathrm{d}}| \times |\mathcal{E}_{\mathrm{d}}|}$ is defined as*

$$\forall\ (ij), (kl) \in \mathcal{E}_{\mathrm{d}} \qquad B_{(ij)(kl)} = \delta_{jk}(1 - \delta_{il}).$$

Note that on a undirected graph, A is symmetric, while B is not.

Remark 1.1. *In Definition 1.8 the entries of the matrix B are in one-to-one correspondence with the edges of a* directed *graph $\mathcal{G}(\mathcal{V}, \mathcal{E}_{\mathrm{d}})$. If the input graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is* undirected*, the non-backtracking matrix is still defined on the set of its directed edges $\mathcal{E}_{\mathrm{d}}$ with $|\mathcal{E}_{\mathrm{d}}| = 2|\mathcal{E}|$ in which the edges (ij) and (ji) are distinguished.*

With the main definitions laid out, we now move to the description of some generative models of random graphs of utmost importance in the remainder of the manuscript. We in particular introduce the Erdős-Rényi model as the simplest random graph generative model, together with its variations adapted to generate random graphs with a community structure.

1.3 RANDOM GRAPHS

In order to get a deeper grasp, make theoretical predictions and have a more profound understanding over *classes of graphs* (and not single realizations) a reductionist approach is needed. Random generative model, depending on few, highly interpretable parameters, are introduced to create graphs that resemble – at least under some aspects – real-world graphs. For a given generative model, then, the probability of an event on the graph (like, *e.g.*, the existence of a cycle of a given length) can be studied. From now on, unless otherwise specified, we considered *undirected* graphs.

"All models are wrong some are useful"
George Box

The reductionist approach has its strengths and weaknesses. Having a well defined probabilistic model to generate $\mathcal{G}(\mathcal{V}, \mathcal{E})$ introduces good mathematical control over different graph-related quantities, particularly in the asymptotic regime of large graphs (with $n \to \infty$). On the down side, of course, simplifying the description of complicated objects into simple generative models leaves necessarily some relevant features behind. For this reason, in the following sections, when introducing the generative models under analysis, we will try to emphasize which property of typical real graphs are caught by the considered generative model and which ones are not.

Many random graphs models have been introduced in the literature. Some relevant examples that will not be discussed since they go beyond the scope of this introduction, are the small world model of [WS98], the preferential attachment model of [AB02] and the Dorogovtsev-Mendes model [DM02]. The remainder will focus on Erdős-Rényi random graphs, together with its extensions to generate graphs with a community structure: the (degree-corrected) stochastic block model.

1.3.1 ERDŐS-RÉNYI RANDOM GRAPHS

Out of all the models of random graphs, the ER model [ER60] is the simplest and most studied. The ER graphs constitute a class of unweighted and undirected graphs. The essence of the ER model is to generate the edges of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ independently at random with equal probability. Formally, recalling the bijection between $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and the adjacency matrix associated to it, we define the ER model as follows:

Definition 1.9 (Erdős-Rényi graph)**.** *Let $c \in \mathbb{R}^+$ be a positive scalar. The entries of the adjacency matrix $A_{ji} = A_{ij}$ of a random Erdős-Rényi graph are set to 1 independently at random with probability $\frac{c}{n}$ and to 0 otherwise.*

Emergence of a giant component

From a straightforward calculation, one obtains for $n \gg 1$ that c is the expected average degree of the graph. The ER random graphs show some relevant properties, especially in the asymptotic regime of $n \to \infty$ that will be assumed in the following. The first of these properties concerns the *percolation threshold*, that is, establishing for what c the largest connected component of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ has a number of nodes proportional to n. This largest connected component takes the name, in the asymptotic regime, of *giant component* and there exists a sharp transition in the value of c determining its appearance [BJR07], as shown in Figure 1.5.

Property 1.1 (Percolation threshold for ER)**.** *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an Erdős-Rényi random graph as in Definition 1.9. Then, for all large n with high probability, the largest component of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ has $O_n(n)$ nodes if and only if $c > 1$. If it exists, there is only one connected component with $O_n(n)$ nodes.*

The notation $a_n = O_n(b_n)$ is equivalent to $\lim_{n\to\infty} b_n/a_n = \kappa$, with κ independent of n. The percolation threshold has applications in solid state physics, epidemiology and technological networks among others [KNZ14].

We now take into consideration the degree distribution of ER random graphs [BB01, Chapter 3].

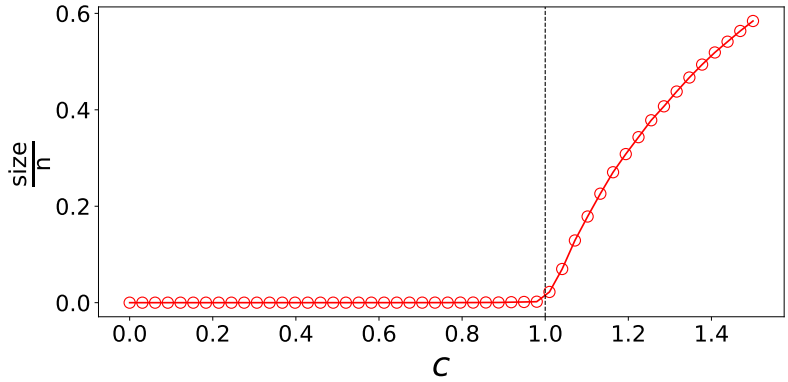
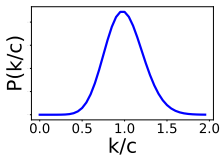


Figure 1.5: Size of the largest connected component divided by n as function of c for different ER-generated random graphs. The black vertical line is at $c = 1$. For this simulation, $n = 10^6$ and no averaging has been taken.

Property 1.2 (Degree distribution of ER). *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a Erdős-Rényi random graph as in Definition 1.9 with $n \to \infty$. Then, the following facts hold*

The degree distribution of ER random graphs



i) The probability distribution of the degree sequence follows a Binomial distribution. Let $p = c/n$

$$\mathbb{P}(d_i = k) = \binom{n-1}{k} p^k (1-p)^{n-1-k},$$

hence in the large n limit, the expected average degree is equal to c.

ii) If $c/\log(n) \to \infty$ as $n \to \infty$, then, with high probability

$$\max_{i \in \mathcal{V}} |d_i - c| = o_n(c).$$

iii) For $n \to \infty$, with high probability,

$$\min d_i > 0 \iff c > \log(n).$$

The notation $a_n = o_n(b_n)$ is equivalent to $\lim_{n\to\infty} a_n/b_n = 0$. From Property 1.2, it turns out that for a sufficiently high average degree c, the degree distribution is almost regular, hence far from the properties of degree heterogeneity studied in real networks. As a matter of fact, for small values of c, even if the degree distribution is not almost regular, it is far from being a broad scale free distribution. As a consequence, the degree distribution of ER random graphs is not suited to describe real-world graphs. Property 1.2 further states the existence of a *connectivity transition*: below the critical threshold, $\mathcal{G}(\mathcal{V}, \mathcal{E})$ has isolated nodes, while above the threshold it is connected with high probability.

Among the properties typically observed in real world networks, we recall the *small world effect* according to which there exist a short path (compared to n) connecting any two nodes in $\mathcal{G}(\mathcal{V}, \mathcal{E})$. This property holds with high

Small world effect in ER graphs

probability also in ER graphs [ABo2, for example], if $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is connected. This means that ER random graphs are a suitable candidate to model small-world effect of real-world graphs.

Property 1.3 (Average distance between nodes of ER)**.** *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be an Erdős-Rényi random graph as in Definition 1.9, with $c > \log(n)$. The average distance between two arbitrarily chosen nodes is proportional to $\log(n)/\log(c)$.*

For $1 < c < \log(n)$, the presence of disconnected nodes is such that the distance between two arbitrarily chosen nodes diverges but Property 1.3 can be extended to this case considering only the giant component of $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

The asymptotic definition of sparse *and* dense *graphs*

With these properties of the degree distribution being laid out, we take the occasion to formally introduce the concept of *sparse* and *dense* graphs in the asymptotic regime. An ER graph is said to be *sparse* if, in the asymptotic limit of large n, its average degree stays finite, *i.e.* if $c = O_n(1)$. Conversely, if the expected average degree goes to infinity, the graph is said to be *dense*. It is worth making further considerations on the asymptotic definition of sparsity and relate it with the intuitive one introduced in Section 1.1. In that context, *sparse* indicated a network with an average degree much smaller than its size n, while here a graph is *sparse* if its average degree stays finite as n goes to infinity. The two ideas are somewhat related, but they are not coincident. From a practical viewpoint, all networks are finite and one could question the efficacy of a definition that only holds in the asymptotic regime. How to state for finite n whether a network is sparse or not?

On the asymptotic definition of sparsity vs the network-related concept of sparsity

The concept of sparsity (and of density) in the asymptotic limit is much more related to *random matrix theory* (RMT) than it is to network theory. It has nonetheless to be noted that also the asymptotic definition of sparsity is not unique, with some authors (see *e.g.* [BGBK+20]) that would define sparse a graph in which $c = o_n(\log(n))$. In agreement with some relevant works on CD on sparse graphs [Dec+11; Krz+13; BLM15] and the fact that RMT naturally comes into play in our work, we adopt the asymptotic definition of sparsity, imposing $c = O_n(1)$. We will explain however that this choice is not compromising, showing that our proposed algorithms do not rely on the specific definition of sparsity.

Locally tree-like structure

To conclude this section concerning the main properties of ER random graphs, we introduce the locally tree-like structure that characterizes the sparse regime. Let us denote with $\mathcal{G}_i(\mathcal{V}, \mathcal{E})$ a *rooted graph*, *i.e.* a graph in which a particular node $i \in \mathcal{V}$ (the *root*) is specified. Denote with $\mathfrak{B}_i(t)$ the ball of radius t around the node i, *i.e.* the sub-graph made by the set of all nodes that can be reached from i in at most t steps and the corresponding edges. If the law of $\mathfrak{B}_i(t)$ under uniformly random sampling of the root admits a limit $\mathcal{L}$, then we call it *local weak limit* [Sal11]. In words, $\mathcal{L}$ is the asymptotic local distribution of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ as seen from a random vertex. For a more formal statement of the concept of local weak convergence, refer to [Sal11; DM+10a]. For a *sparse* ER random graph, with $c = O_n(1)$, the law of a uniformly randomly rooted ER converges locally to the Poisson *Galton-Watson* (GW) distribution with parameter c.

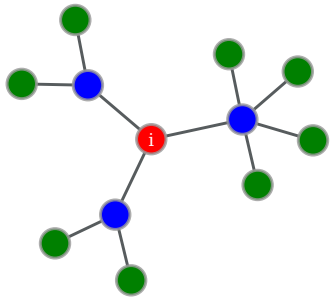


Figure 1.6: A toy example of a Poisson GW tree. In red the root i, in blue the first generation of nodes, in green the second.

Property 1.4 (Convergence to Poisson Galton-Watson tree of ER)**.** *A sparse ER random graph rooted at i with $c = O_n(1)$ converges locally to a Poisson GW tree so obtained: consider the node i as the root and generate d_i neighbours (called* sons*), where d_i is a Poisson[2] random variable with parameter c and iteratively repeat the operation for each son, creating the following generations.*

As a consequence of this property, a sparse ER graph locally *looks like* a tree and hence, with high probability, there are no cycles of finite size. Figure 1.6 displays an example of a Poisson GW tree, rooted at i.

Real networks are sparse but not tree-like

The tree-like approximation of sparse graphs will be of fundamental use in the following but it raises an important question when compared to real-world graphs. We argued earlier that real networks are sparse in the sense that the average degree is very small compared to n and that it is convenient to introduce an asymptotic concept of sparsity in prevision to the applications to RMT. As a "side effect" however, the resulting ER graph turns out to be locally tree-like. This property is not necessarily met in real networks in which, on the opposite, many short cycles may be present. We will be oblivious of this major difference for most of the follow up discussions in which the tree-like assumption will be heavily exploited. This question will reappear in Chapter 7 in which we will consider the validity of our theoretical approach (based on the tree-like approximation) on real graphs.

The main feature captured by the ER random model is the small world effect which is of major importance in complex networks. As we mentioned already in Section 1.1, the focus of our work is devoted to the problem of node classification hence on how the nodes of a graph can be divided into groups, according to the preferential attachment described by the edge configuration[3]. The ER model completely misses this point, since all nodes are "equal" and no reasonable partition can be made. It can rather be seen as good null model: any algorithm should not find a community structure on

2 Note that in the sparse regime, for $n \to \infty$ the Binomial distribution convergences towards a Poisson distribution.

3 A different possible way to encode information concerning the nodes' features, is by assigning a weight to each edge. This case is, for the moment, not considered and its analysis is deferred to Chapter 10

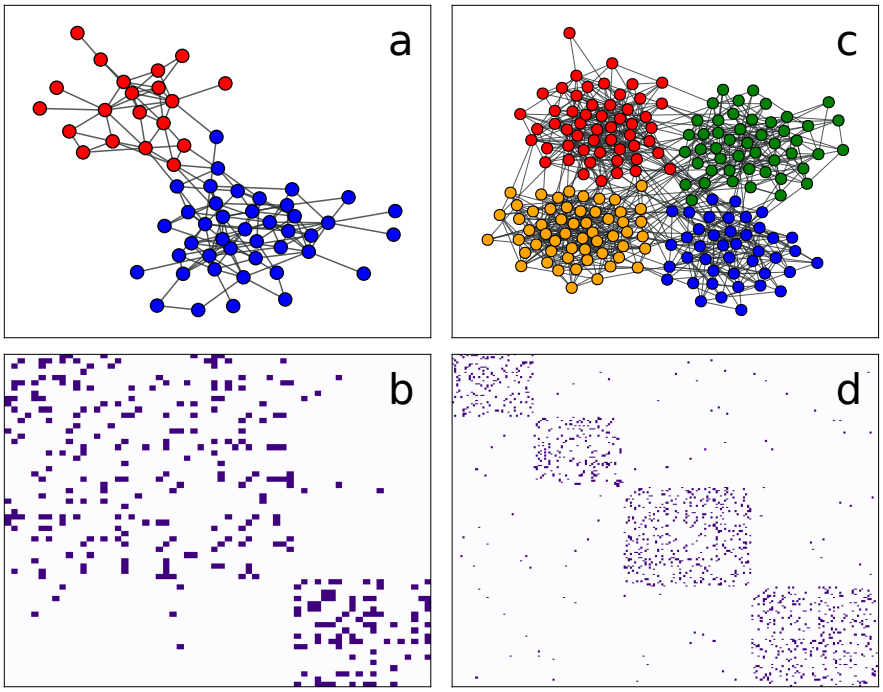


Figure 1.7: **a**: the *dolphin* network [Lus03] with the colour code indicating class membership. **b**: the adjacency matrix of *a*. **c**: a realization of the SBM (Definition 1.10) with $k = 4$ communities with colour code indicating the class membership. **d**: the adjacency matrix of *c*.

a ER random graph. We now proceed introducing two variations to the ER model that describe random graphs with a community structure.

1.3.2 THE STOCHASTIC BLOCK MODEL

The community structure of real graphs

As a consequence of the homophilic nature of interactions on social graphs, typically real networks are structured in communities [GN02]. A more extensive discussion on communities and the problem of community detection will be treated in Part II. As an example, the plot **a** of Figure 1.7 displays a real network with a community structure: the node colour code identifies the communities and clearly evidences that connections within communities are more likely than connections across communities. This can be equivalently visualized in terms of the matrix A, as shown in the plot **b** of Figure 1.7, in which the basis of the adjacency matrix is chosen so that nodes in the same community are close to one another.

The complete homogeneity of ER random graphs makes them unsuited to describe this self-organization behavior and a more involved model is necessary. We introduce a variation of the ER model, which goes under the name of *stochastic block model* (SBM) [HLL83].

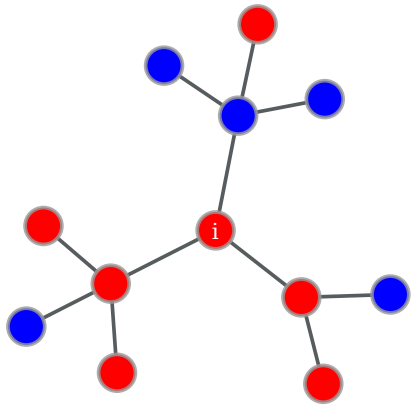


Figure 1.8: A toy example of a Poisson GW tree with a two class community structure. The color code indicates the class label assignment.

The stochastic block model

Definition 1.10 (Stochastic block model). *Let $\boldsymbol{\ell} \in \{1, \dots, k\}^n$ be the class label vector, where k is the number of classes and $\mathbb{P}(\ell_i = a) = \pi_a$. Further let $C \in \mathbb{R}^{k \times k}$ be a symmetric matrix with positive elements. The entries of the matrix $A_{ij} = A_{ji}$ are set to one independently at random with probability*

$$\mathbb{P}(A_{ij} = 1) = \min\left(\frac{C_{\ell_i, \ell_j}}{n}, 1\right),$$

and are equal to zero otherwise.

The matrix C represents the class affinity matrix and imposes different average degrees between nodes in the same community and nodes in different communities, hence reproducing the class structure. The vector $\boldsymbol{\pi} = (\pi_1, \dots, \pi_k)$ determines the class sizes. Let $\mathcal{V}_a = \{i :\in \mathcal{V} \ : \ \ell_i = a\}$, then it is easy to see that the expectation of the class size is $\mathbb{E}[|\mathcal{V}_a|] = n\pi_a$. Plot **c** of Figure 1.7 shows a realization of a graph generated from the SBM, while the plot **d** displays the corresponding adjacency matrix.

The SBM inherits many of the properties of the ER model: the small world effect, the degree distribution and the convergence to a Poisson GW tree in the sparse regime. In particular, we now define a GW tree with labels: for each node with label a, a Poisson random number with mean $C_{ab}\pi_b$ of sons will be assigned to class b [MNS18, for istance]. For example, in the case of two communities of equal size ($\boldsymbol{\pi} = \mathbf{1}_2/2$), the expected number of neighbours of a node in class a is C_{aa} nodes in class a, and C_{ab} nodes in class b, with $C_{aa} > C_{ab}$. A toy representation of this process is shown in Figure 1.8.

As we just mentioned, the SBM, like the ER, has a very regular degree distribution, not adapted to model real-world networks. To fix this limitation, the *degree corrected stochastic block model* (DCSBM) [KN11] is introduced which, in addition to generating a class structure also allows one to have an arbitrary node degree distribution.

The degree corrected stochastic block model

Definition 1.11 (Degree corrected stochastic block model). *Let $\boldsymbol{\ell} \in \{1,\dots,k\}^n$ be the class label vector, where k is the number of classes and $\mathbb{P}(\ell_i = a) = \pi_a$ and let $C \in \mathbb{R}^{k\times k}$ be a symmetric matrix with positive elements.*

Let $\theta \in \Theta = [\theta_{\min}, \theta_{\max}]$ be a random variable that encodes the intrinsic node connectivity, distributed according to ν, satisfying $\int_\Theta d\nu(\theta) = 1$ (normalization), $\mathbb{E}[\theta] = \int_\Theta \theta\, d\nu(\theta) = 1$, $\mathbb{E}[\theta^2] = \int_\Theta \theta^2\, d\nu(\theta) \equiv \Phi = O_n(1)$. For each node, θ_i is drawn independently at random from the distribution ν.

The entries of the matrix $A_{ij} = A_{ji}$ are set to one independently at random with probability

$$\mathbb{P}(A_{ij} = 1) = \min\left(\theta_i\theta_j \frac{C_{\ell_i,\ell_j}}{n}, 1\right),$$

and are equal to zero otherwise.

From a simple computation, one can see that the expected average degree of node i is proportional to θ_i, *i.e.* $\mathbb{E}[d_i] \propto \theta_i$. Consequently the vector $\boldsymbol{\theta} = (\theta_1,\dots,\theta_n)$ can be used to produce any degree distribution on the graph.

Percolation threshold in the DCSBM

For the DCSBM we established in [DCT20a] the percolation threshold in the particular case in which the expected average degree is the same for all classes. The result is formally stated as follows:

Theorem 1.1 (Percolation threshold in DCSBM, [DCT20a]). *Consider a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ generated according to the DCSBM procedure as in Definition 1.11. Let $\Pi = \mathrm{diag}(\pi_1,\dots,\pi_k)$ and assume that the constant c is the largest eigenvalue of $C\Pi$ with eigenvector $\mathbf{1}_k$. Then, for all large n, with high probability, $\mathcal{G}(\mathcal{V},\mathcal{E})$ has a giant component if and only if $c\Phi > 1$.*

From a simple calculation, one obtains that c is the expected average degree of $\mathcal{G}(\mathcal{V},\mathcal{E})$ and Theorem 1.1 can be straightforwardly related to Property 1.1. The proof of Theorem 1.1 is provided in Appendix A.

To conclude this Section, let us recapitulate the main properties of the DCSBM. The DCSBM captures the small world effect typically observed in real networks and is suited to produce a graph with an arbitrary degree distribution (hence even a broad, power law distribution). The typical organization into communities of real networks is also correctly described by the DCSBM. This model however inherits, in its sparse version for $C_{ab} = O_n(1)$, a locally tree-like structure[4] from the ER and the SBM. As we mentioned already, this property will be heavily exploited in the remainder and the legitimacy of ignoring this discrepancy will be further discussed in Chapter 7.

This concludes our introduction to graphs, leaving place to the next chapter on statistical physics. In particular, we will introduce the Ising model on a graph, in which to each node is associated a random variable s_i called *spin*.

4 This time the size of each labelled progeny is a Poisson with mean $C_{ab}\pi_b\Phi$ [GLM15].

2

THE ISING MODEL ON SPARSE GRAPHS

Abstract

This chapter provides an introduction to statistical physics, with a particular attention to the Ising model describing interacting spins placed in correspondence to the nodes of a graph. This model is studied with the help of two closely related approximations (namely belief propagation *(BP) and the Bethe approximation) that are asymptotically exact on locally tree-like graphs such as the sparse realizations of the* degree corrected stochastic block model *(DCSBM). The tools introduced in this chapter will be extensively exploited in Parts II, III.*

Statistical physics plays an important role in shaping the technical parts of this manuscript in terms of methods, approximations and intuitions. In fact, statistical physics has significantly contributed to the analysis and design of new (and old) algorithms for machine learning and, more in general, for statistical inference [Car+19; Iba99; ZK16; OS01; MM09, for instance]. Specifically, parts II and III show how *Ising-like* models [Isi25] on graphs can be exploited to interpret and improve efficient algorithms for *spectral clustering* (SC). The role of this chapter is hence to provide the reader a basic description of the methods, approximations and interpretations of Ising-like models on graphs.

2.1 THE BOLTZMANN DISTRIBUTION

Statistical physics has its origin back in the second half of 19^{th} century when a lot of effort was conveyed by the scientific community to give a mechan-

ical interpretation to thermal phenomena [Bru76a]. The laws of thermodynamics describe the behavior of (almost infinitely) many bodies (particles, molecules...) from a coarse grained perspective, with only few parameters such as the *pressure* and the *temperature*. These laws rely on axioms that do not invoke any mechanical principle, using instead the concept of *heat*. Although not formally defined, back in the 19^{th} century, there was a general agreement that heat could be intended as a form of movement [Bru76b].

The example of the free gas

As a historically relevant example [Max67], let us consider the classical *free gas* problem to provide a practical case-study. With Newton laws it is in principle possible, given a model describing the gas molecules interactions, to compute the trajectories of each molecule across time. In practice, it turns out that this is more of a mental experiment than other, since the typical number of molecules inside an average size room is of order 10^{23}: for this magnitude of the system, any analytical or numerical solution of Newton's equations is unapproachable, even with powerful modern computers. To the impossibility of solving such a large system of equations, one needs to add that of *measuring* the initial conditions of the system (*i.e.* positions and velocities of all 10^{23} particles) needed to solve Newton's equations.

A *mechanical* study of the free gas seems consequently unapproachable. According to thermodynamics instead, only few parameters are sufficient to describe the free gas: the pressure, temperature, volume and number of molecules [Cla34]. This suggests that the information encoded by the trajectories is redundant and way too rich: many different gas configurations lead to the same measurable macroscopic quantities.

From trajectories to probability distributions

This evidence required a new perspective relating the microscopic (where Newton's law are defined) to the macroscopic (where thermodynamics lays) worlds. It was Boltzmann who gave the main contributions in the outbreak of the probabilistic interpretation of thermodynamic [Gib14]. In this picture, the system as a whole is considered from a statistical viewpoint, and the goal is determining the probability distribution (rather than the exact values) of the position and velocity of molecules. This way, only few, measurable parameters are sufficient to describe the system, relating the microscopic scale interpretation to macroscopic thermodynamics observables [Par88].

Let us now formalize the probabilistic approach introduced by Boltzmann, first defining the *phase space*. Let n be the number of items composing the system and let d_{f} be the number of degrees of freedom each item has. In the free gas example, n is the number of molecules and $d_{\text{f}} = 6$ are the three coordinates of space and the three coordinates of velocity. Each gas configuration is encoded by the vector $\boldsymbol{f} = \{f_1^1, \dots, f_{d_{\text{f}}}^1, \dots, f_1^n, \dots, f_{d_{\text{f}}}^n\}$ containing all the coordinates of all degrees of freedom of each item. The space Ω in which $\boldsymbol{f}$ lives is called the *phase space*. A configuration of the system corresponds to a point in Ω and its evolution to a trajectory in Ω. Supposing

to be at thermal equilibrium,[1] each configuration $f \in \Omega$ is visited with the following probability

$$\mu_\beta(f) = \frac{e^{-\beta \mathcal{H}(f)}}{Z_\beta}. \tag{2.1}$$

The Boltzmann distribution

This is the so-called *Boltzmann distribution*. The parameter $\beta = (\kappa_B T)^{-1}$ is the inverse temperature,[2] where κ_B is the Boltzmann constant and T the temperature of the system. The function $\mathcal{H}(f)$ is called *Hamiltonian* and corresponds to the energetic cost of each configuration f. Finally Z_β is the normalization constant that, assuming that the variables f live in a discrete space,[3] is written as follows

$$Z_\beta = \sum_{f \in \Omega} e^{-\beta \mathcal{H}(f)}.$$

Averages over the Boltzmann distribution (2.1) are denoted with $\langle \cdot \rangle_\beta$. Taking a closer look at Equation (2.1), each configuration $f \in \Omega$ is visited with a probability that depends on its energetic cost, favouring small energetic configurations and its expression gives a microscopic interpretation of the system. Three facts should be underlined at this point:

- all configurations with finite energy can be visited with non-vanishing probability;
- the temperature determines the rate of exploration: for large temperatures (small β) high energy configurations are more likely to be visited;
- each observable quantity is a function of the system configuration.

Now that the main concepts have been laid out, we define and study the Ising model [Isi25] which was developed to provide a microscopic interpretation to spontaneous magnetization and is one of the most studied and well understood problems in statistical physics.

2.2 THE ISING MODEL

Some materials, called *ferromagnetic*, share the peculiar property of having a spontaneous magnetic behaviour in absence of an external magnetic field when taken below a critical, material dependent, temperature T_c, also known

1 The thermal equilibrium corresponds to a condition of the system such that the observables are not evolving with time.

2 Note that β has the dimensionality of an inverse energy, but for simplicity we refer to it as *inverse temperature*.

3 This assumption is made for simplicity as all the results can be reformulated in the continuous limit using integrals instead of sums. In the following all considered configurations spaces are indeed going to be discrete, justifying the reason of this choice.

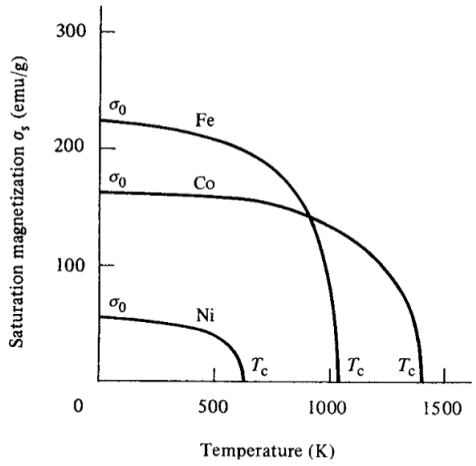


Figure 2.1: Saturation magnetization of iron (Fe), cobalt (Co) and nickel (Ni) as a function of the temperature. Picture taken from [CG11].

as *Curie temperature* [CG11]. Above T_c, instead, no magnetic property is observed. This behavior defines a *phase transition* between the *ferromagnetic phase* at $T < T_c$ in which the magnetic behavior is observed and the *paramagnetic phase* at $T > T_c$ in which no magnetic behaviour exists. An experimental plot of this phase transition is shown in Figure 2.1

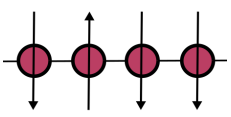
The ambition of the Ising model is to provide a microscopic interpretation of the origin of spontaneous magnetization. The idea is to suppose that a ferromagnetic material is composed by a large number of interacting dipoles, each one with its own magnetization, called *spins*, $s_i \in \{\pm 1\}$. The magnetization of the material is given by the sum of all these spins: if they are randomly arranged, then no magnetization appears, while it does otherwise. Referring to Section 2.1, in this case the feature vector is $\boldsymbol{f} \equiv \boldsymbol{s}$ and the phase space is $\Omega = \{\pm 1\}^n$. The summation is denoted $\sum_{s\in\Omega} \equiv \sum_{\boldsymbol{s}}$ for convenience. The Hamiltonian, appearing in the Boltzmann distribution (2.1), reads for the Ising model[4] on an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$

The Ising Hamiltonian

$$\mathcal{H}(\boldsymbol{s}) = -\sum_{(ij)\in\mathcal{E}} J_{ij} s_i s_j - \sum_{i\in\mathcal{V}} h_i s_i, \tag{2.2}$$

where the term J_{ij} models the interaction between the spins s_i and s_j and h_i is a local magnetic field applied to the spin s_i. Equation (2.2) is actually the most generic form of the Hamiltonian: the problem originally solved by Ising considers a one dimensional spin chains with:

Four spins of a 1d Ising chain



$$h_i = h \quad \forall\, i \tag{2.3}$$

$$J_{ij} = \mathrm{J}\left(\delta_{j,i+1} + \delta_{j,i-1}\right), \tag{2.4}$$

for $\mathrm{J} > 0$. According to Equations (2.2, 2.4) there is an energetic gain for neighbouring spins to be aligned rewarding them with a negative energy

4 The expression "Ising Hamiltonian" for arbitrary couplings and fields is a slight abuse of language that is made to indicate that the spins are $s_i \in \{\pm\}$.

contribution, while penalizing neighbouring spins with opposite orientation with a positive energy contribution. The external field h polarizes all spins towards the same direction. The lowest energy configuration of the Hamiltonian (2.2) (called the *ground state*) under the constraint (2.4) is hence obtained for $\boldsymbol{s} = \mathbf{1}_n$, corresponding to the *ferromagnetic configuration* in which all spins are aligned.

To determine whether or not this model predicts spontaneous magnetization, it is necessary to study the explicit expression of the magnetization $\boldsymbol{m} = \langle \boldsymbol{s} \rangle_\beta$. Sadly for Ising, in absence of external magnetic field ($h = 0$), no magnetization appears in one dimension. The model can however explain spontaneous magnetization in dimension greater or equal than two.[5] To obtain the analytical expression of $\boldsymbol{m}$ it is necessary to compute the dimensionless free energy F_β, defined as follows.

Definition 2.1 (Dimensionless free energy)**.** *Let Z_β be the normalization constant appearing in Equation (2.1). The dimensionless free energy is*

$$F_\beta = -\log Z_\beta.$$

$\log(Z_\beta)$ *is a moment generating function of the Boltzmann distribution*

The term *dimensionless* comes from the fact that F_β, so defined does not have the physical dimensionality of an energy. The magnetization is obtained taking a derivative of F_β with respect to the external field:

$$\begin{aligned} m_i = \langle s_i \rangle_\beta &= \frac{1}{Z_\beta} \sum_{\boldsymbol{s}} s_i e^{-\beta \mathcal{H}(\boldsymbol{s})} = \frac{1}{\beta Z_\beta} \partial_{h_i} \sum_{\boldsymbol{s}} e^{-\beta \mathcal{H}(\boldsymbol{s})} \\ &= \frac{1}{\beta Z_\beta} \partial_{h_i} Z_\beta = \frac{1}{\beta} \partial_{h_i} \log Z_\beta = -\frac{1}{\beta} \partial_{h_i} F_\beta. \end{aligned}$$

So, computing analytically the free energy corresponds to the main goal and it is this function that determines the state of the system. The energy F_β can alternatively be written in a form known from classical thermodynamics. Denoting the average energy with $U = \sum_{\boldsymbol{s}} \mathcal{H}(\boldsymbol{s}) \mu_\beta(\boldsymbol{s})$ and the entropy with $S = -\kappa_B \sum_{\boldsymbol{s}} \mu_\beta(\boldsymbol{s}) \log \mu_\beta(\boldsymbol{s})$, then, from a straightforward calculation

$$F_\beta = \beta(U - TS).$$

At low T, the configurations with small energy are favoured (see also Equation (2.1)), while as the temperature increases, the entropic term gains more weight and high energy configurations are more likely to be observed. The temperature balances these two terms causing a phase transition between the ordered ferromagnetic configuration at low T to a disordered paramagnetic configuration at high T.

5 Strictly speaking, spontaneous magnetization is observed also for a dimensionality $d \in (1, 2)$. A non-integer value d indicates a fractal dimensionality [Vez03, for instance]

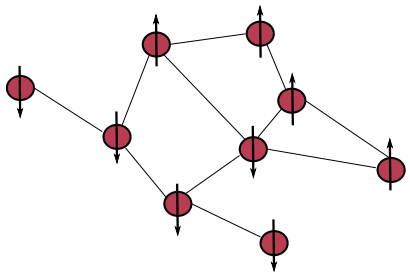


Figure 2.2: A toy model of the Ising model on a graph. The arrows indicate the spins orientation.

The whole problem of computing $\boldsymbol{m}$ and consequently also T_c hence boils down to finding an explicit expression to Z_β. Unfortunately, this problem can be solved for only few choices of J, such as (for instance) those corresponding to the 1-dimensional Ising chain [Isi25], the 2-dimensional square lattice [Ons44], the fully connected Curie-Weiss model [KPW13], the Bethe lattice [Tho86]. When more involved geometries intervene and J corresponds to the (weighted) adjacency matrix on an arbitrary graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, the exact computation of the partition function Z_β becomes intractable [DGM08] and approximate methods must be adopted to obtain the result.

The interest in studying the Ising model for more complex geometries and interactions comes from the fact that it represents the perhaps most elementary way to model pairwise interactions. Its applicability is consequently not bounded to the explication of spontaneous magnetization, but has found several applications in a variety of fields ranging from optimization problems, computer science, biology, sociology and economics [CFL09; SN13]. For this reason, we consider in the remainder the problem of how to determine whether or not the system is in the paramagnetic phase[6] in the case in which J is a weighted adjacency matrix of a graph. The focus will be on two methods, namely the *Bethe approximation* and the BP algorithm, to study the Ising model on graphs.

From now on we assume that J is the weighted adjacency matrix of a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (as sketched in Figure 2.2) with $h_i = 0$ for all i.

6 In the case of a ferromagnetic Ising Hamiltonian in which $J_{ij} = J > 0$ for all $(ij) \in \mathcal{E}$, only two phases exist: the paramagnetic and the ferromagnetic. In this case, determining the stability condition of the paramagnetic phase is equivalent to determining the Curie temperature. For other choices of J, the paramagnetic phase always exists, but the ferromagnetic phase does not necessarily appear and other phases may emerge, instead. This case will be further considered in Section 3.4 and in Chapter 10.

2.3 TREE-LIKE APPROXIMATIONS

In this section we introduce two approximate methods to study Ising-like models on tree-like weighted graphs. Both approximations are introduced in a discrete phase space more general than $\Omega = \{\pm 1\}^n$, but deeper considerations on the model and methods on how to determine the emergence of the paramagnetic phase are made for Ising variables.

The two approximations here considered are the *cavity method*, also known in the computer science literature under the name of *belief propagation* (BP) [MPV87; Pea14; YFW+00] and the variational Bethe approximation [Bet35; Pei36]. The two methods are adapted to sparse, locally tree-like graphs and introduce the non-backtracking and Bethe-Hessian matrices, of central importance to Parts II and III, providing them with a physical interpretation.

2.3.1 BELIEF PROPAGATION

Belief propagation is an iterative algorithm used to compute the marginal probabilities of a factorized probability distribution, such as $\mu_\beta(\boldsymbol{s})$. Lemma 2.1 in particular is the main result on which BP algorithm builds, providing an exact expression of the node and edge marginals of a probability distribution defined on a tree [WJ08, for instance].

Lemma 2.1 (Belief propagation on a tree). *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a tree and let $\mu(\boldsymbol{s})$ be a probability distribution defined on $\mathcal{G}(\mathcal{V}, \mathcal{E})$ satisfying*

$$\mu(\boldsymbol{s}) = \prod_{(ij)\in\mathcal{E}} \phi_{ij}(s_i s_j), \tag{2.5}$$

with $\boldsymbol{s} \in \{1, \dots, k\}^n$. Then the edge marginal $\mu_{ij}(s_i, s_j) = \sum_{\boldsymbol{s}\setminus s_i, s_j} \mu(\boldsymbol{s})$ and the node marginal $\mu_i(s_i) = \sum_{\boldsymbol{s}\setminus s_i} \mu(\boldsymbol{s})$ can be written in the following form:

$$\mu_i(s_i) = \prod_{k\in\partial i} \eta_{ki}(s_i) \tag{2.6}$$

$$\mu_{ij}(s_i s_j) = \phi_{ij}(s_i s_j) \prod_{k\in\partial i\setminus j} \eta_{ki}(s_i) \prod_{l\in\partial j\setminus i} \eta_{lj}(s_j). \tag{2.7}$$

The quantities $\eta_{ij}(s_j)$ are defined on the set of directed edges $\mathcal{E}_{\mathrm{d}}$ and are called messages.

From Equation (2.6, 2.7), exploiting $\mu_i(s_i) = \sum_{s_j} \mu_{ij}(s_i, s_j)$, it is obtained that the *messages* have to satisfy the following fixed point equation

$$\eta_{ji}(s_i) = \sum_{s_j} \phi_{ij}(s_i, s_j) \prod_{l\in\partial j\setminus i} \eta_{lj}(s_j). \tag{2.8}$$

Message passing

Solving Equation (2.8) and then injecting the result in Equation (2.6, 2.7), one obtains the node and edge marginal distributions. Of course, Lemma 2.1

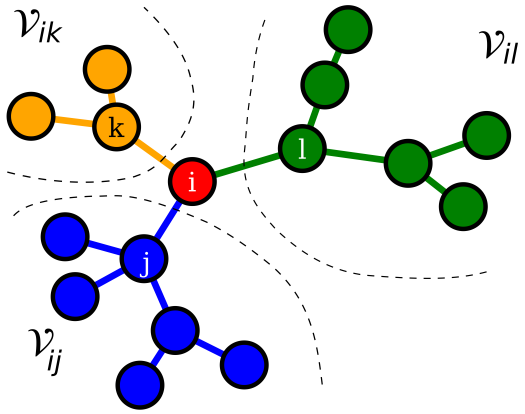


Figure 2.3: Sketch of a tree. The node i in red, while in green, blue and orange the edges and nodes $\mathcal{E}_{ix}, \mathcal{V}_{ix}$ with $x = j, k, l$, respectively.

is relevant for determining the magnetization of the Ising model on a tree, since the Boltzmann distribution $\mu_\beta(\boldsymbol{s})$ can precisely be written in the form of Equation (2.5). Lemma 2.1 however only applies to trees and not to graphs with cycles for which *loopy* BP should be instead considered. Before detailing the extension of BP to the case of graphs with cycles (hence that are not trees) we sketch here the proof of Lemma 2.1 since it is very pedagogical and helpful to understand the essence of BP algorithm.

Proof of Lemma 2.1. Consider an directed edge $(ij) \in \mathcal{E}_{\mathrm{d}}$ and let $\mathcal{E}_{ij}$ be defined as follows

$$\mathcal{E}_{ij} = \{(kl) \in \mathcal{E} \ : \ \exists \, P^{\mathrm{NB}}_{(ij),(kl)} \wedge P^{\mathrm{NB}}_{(ij),(lk)}\},$$

where $\mathcal{P}^{\mathrm{NB}}_{e,e'}$ is a non-backtracking path from for e to e'. By convention, if no such path exists, then $\mathcal{P}^{\mathrm{NB}}_{e,e'} = \emptyset$, while $(ij) \in \mathcal{E}_{ij}$. In words, $\mathcal{E}_{ij}$ is the set of all undirected edges that can be reached from i only passing through j. As a consequence of the fact that on a tree there exists a unique path connecting any two nodes, the two following properties are verified:

$$\forall\, i \in \mathcal{V}, \quad \mathcal{E} = \bigcup_{k \in \partial i} \mathcal{E}_{ik}\,; \tag{2.9}$$

$$\forall\, (ij) \in \mathcal{E}, \quad \mathcal{E} = \{(ij)\} \cup \underbrace{\bigcup_{k \in \partial i \setminus j} \mathcal{E}_{ik}}_{\text{reached from } (ji)} \cup \underbrace{\bigcup_{l \in \partial j \setminus i} \mathcal{E}_{jl}}_{\text{reached from } (ij)}. \tag{2.10}$$

Furthermore, note that $\forall\ j \neq k,\ \mathcal{E}_{ij} \cap \mathcal{E}_{ik} = \emptyset$. A pictorial representation of the definition of $\mathcal{E}_{ij}$ is given in Figure 2.3. Exploiting Equation (2.9), $\mu(\boldsymbol{s})$ can then be written as:

$$\mu(\boldsymbol{s}) = \prod_{(ab) \in \mathcal{E}} \phi_{ab}(s_a, s_b) = \prod_{k \in \partial i} \prod_{(ab) \in \mathcal{E}_{ik}} \phi_{ab}(s_a, s_b) \equiv \prod_{k \in \partial i} \psi_{ik}(\boldsymbol{s}_{\mathcal{V}_{ik}}),$$

where $\mathcal{V}_{ik}$ is the set of nodes connected by edges in $\mathcal{E}_{ik}$ (i included). The node marginal can then be written in the following form

$$\mu_i(s_i) = \sum_{\boldsymbol{s}\setminus s_i} \mu(\boldsymbol{s}) = \sum_{\boldsymbol{s}\setminus s_i} \prod_{k\in\partial i} \psi_{ik}(\boldsymbol{s}_{\mathcal{V}_{ik}}) = \prod_{k\in\partial i} \sum_{\boldsymbol{s}_{\mathcal{V}_{ik}}\setminus s_i} \psi_{ik}(\boldsymbol{s}_{\mathcal{V}_{ik}}).$$

Denoting $\eta_{ji}(s_i) = \sum_{\boldsymbol{s}_{\mathcal{V}_{ik}}\setminus s_i} \psi_{ik}(\boldsymbol{s}_{\mathcal{V}_{ik}})$, we obtain the first equation of Lemma 2.1. Note that $\eta_{ji}(s_i)$ indeed only depends on s_i since the sum is run over all spins $\boldsymbol{s}_{\mathcal{V}_{ik}}$, except s_i. Proceeding in a similar way, the expression of the edge marginal is obtained from Equation (2.10).

□

The essence of the proof of Lemma 2.1 relies on the conditional independence of the node variables on trees that we already discussed in Chapter 1. More specifically, to obtain Equation (2.9), one could imagine to remove the node i, obtaining d_i (the degree of i) disconnected sub-graphs in which variables are independent and hence factorize. Similarly Equation (2.10) is obtained removing the nodes i and j from the graph.

The proof of Lemma 2.1 hence heavily exploits the conditional independence and the fact that for any pair of nodes, only a single path connecting them exists, since there are no cycles. But what happens when cycles are present? It was stated in Chapter 1 that sparse ER, SBM, DCSBM graphs are locally tree-like, *i.e.* with high probability they contain cycles but of no finite size. Consequently, removing a node i from $\mathcal{G}(\mathcal{V},\mathcal{E})$ one obtains d_i subsets of nodes that are not independent (since they are not disconnected) but that are only weakly dependent. This intuition is at the basis of *loopy* BP that consists in approximating the marginals in the following way:

BP on graphs with cycles

$$\eta_{ji}(s_i) = \frac{Z_i}{Z_{ji}} \sum_{s_j} \phi_{ij}(s_i,s_j) \prod_{l\in\partial j\setminus i} \eta_{lj}(s_j). \tag{2.11}$$

$$\mu_i(s_i) \approx \frac{1}{Z_i} \prod_{k\in\partial i} \eta_{ki}(s_i) \tag{2.12}$$

$$\mu_{ij}(s_i,s_j) \approx \frac{1}{Z_{ij}} \phi_{ij}(s_i,s_j) \prod_{k\in\partial i\setminus j} \eta_{ki}(s_i) \prod_{l\in\partial j\setminus i} \eta_{lj}(s_j), \tag{2.13}$$

where Z_i, Z_{ij} are the normalization constant and, we recall, the $\approx$ sign comes from the fact that $\mathcal{G}(\mathcal{V},\mathcal{E})$ is not a tree.

The application of the BP algorithm of graphs with cycles is known to give, in general, very good performances [MWJ13], but even more interestingly, it has been shown that on locally tree-like graphs (such as sparse ER) the estimation of the marginals is asymptotically exact [DM+10b]. To confirm this result, Figure 2.4 shows the average magnetization $\bar{\boldsymbol{m}} = \frac{1}{n}\mathbf{1}^T\boldsymbol{m}$ as a function of the inverse temperature β for a ferromagnetic Ising model with $J = JA$, with A the adjacency matrix of a ER sparse graph (Definitions 1.7 and 1.9).

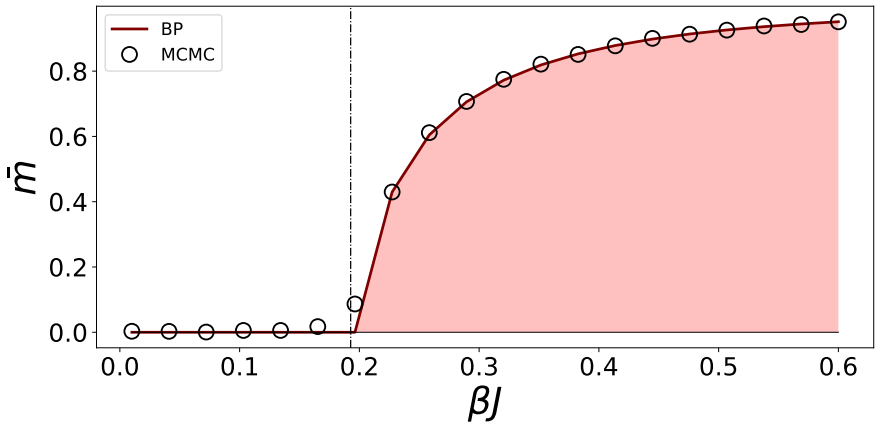


Figure 2.4: Average magnetization as a function of βJ for a Ising model on a graph in which $J = JA$ and A is the adjacency matrix of a ER graph with $n = 30\,000$ and $c = 5$. The red curve is the value of $\bar{\boldsymbol{m}}$ obtained from BP, while the black dots using a Monte Carlo Markov chain simulation. The vertical line is the theoretical prediction of the phase transition.

The prediction of BP is compared to the direct computation of $\bar{\boldsymbol{m}}$ achieved by Monte Carlo method [Has70], evidencing an almost perfect agreement.

High temperature expansion of BP

To conclude this section, we wish to show how the stability of the paramagnetic phase can be estimated from BP on a Ising model (for which we recall that $s_i \in \{\pm 1\}$) and how BP equations are related to the non-backtracking matrix introduced in Definition 1.8. To do so, we consider the so-called *paramagnetic fixed point* of Equation (2.11), corresponding to the case in which the messages are constant, *i.e.*, independent of $\boldsymbol{s}$. This corresponds to the *paramagnetic* phase, since, from Equation (2.12) also the node marginal is independent of $\boldsymbol{s}$, hence $\boldsymbol{m} = \mathbf{0}_n$. Letting $\eta_{ji}(s_i) = \sum_{s_j} \phi_{ij}(s_i, s_j)(1 + \epsilon_{ij} s_j)$, one obtains, to first order in $\epsilon_{(\bullet)}$

$$\epsilon_{ij} = \sum_{l \in \partial j \setminus i} \mathrm{th}(\beta J_{jl}) \epsilon_{jl} + \text{h.o.t.},$$

where h.o.t. denotes the *higher order terms* in $\epsilon_\bullet$. This equation can be more succinctly written as

$$B U_{\beta,J} \boldsymbol{\epsilon} = \boldsymbol{\epsilon} + \text{h.o.t.}, \tag{2.14}$$

where $U_{\beta,J} \in \mathbb{R}^{|\mathcal{E}_\mathrm{d}| \times |\mathcal{E}_\mathrm{d}|}$ has entries $\left(U_{\beta,J}\right)_{e,e'} = \delta_{e,e'} \mathrm{th}(\beta J_e)$ and B is the non-backtracking matrix of Definition 1.8. The critical (inverse) temperature β_c determining the instability of the paramagnetic phase is

$$\lambda_1^{\downarrow|\cdot|}(B U_{\beta_\mathrm{c},J}) = 1,$$

where $\lambda_1^{\downarrow|\cdot|}(\cdot)$ denotes the eigenvalue of a matrix with largest modulus. When $\beta < \beta_\mathrm{c}$, then the paramagnetic phase is stable, since iterating Equation (2.14), the perturbation $\boldsymbol{\epsilon}$ vanishes, while it is not for $\beta > \beta_\mathrm{c}$. The

spectral properties of BU_β are hence fundamental to determine the value of β_c and are considered in Chapter 3.

We now consider the second method adapted to sparse tree-like graphs, namely the *Bethe* approximation.

2.3.2 THE BETHE APPROXIMATION

An alternative method to estimate the averages on factorized probability distribution on sparse graphs relies on a variational approach. We recall that the ultimate goal we have is to obtain an analytic expression of the free energy F_β a task that is, in general, not feasible. The basic idea of variational approaches is to substitute the expression of $\mu_\beta(\boldsymbol{s})$ which makes the problem intractable, with another distribution $p_{\boldsymbol{q}}(\boldsymbol{s})$ that can be treated analytically. The vector $\boldsymbol{q}$ represents a set of parameters on which $p_{\boldsymbol{q}}(\boldsymbol{s})$ depends and $\boldsymbol{q}$ should be chosen so to obtain the best approximation of F_β, given the expression of $p_{\boldsymbol{q}}$. Explicitly this is done introducing the variational free energy $\tilde{F}_\beta(\boldsymbol{q})$ as in Equation (2.15):

$$\tilde{F}_\beta(\boldsymbol{q}) = \sum_{\boldsymbol{s}} p_{\boldsymbol{q}}(\boldsymbol{s}) \left(\beta \mathcal{H}(\boldsymbol{s}) + \log p_{\boldsymbol{q}}(\boldsymbol{s})\right). \tag{2.15}$$

The variational free energy

When can immediately see that, for any distribution $p_{\boldsymbol{q}}(\boldsymbol{s})$, $\tilde{F}_\beta(\boldsymbol{q}) \geq F_\beta$: in fact, exploiting $\beta\mathcal{H}(\boldsymbol{s}) = F_\beta - \log \mu(\boldsymbol{s})$, one obtains

$$\begin{aligned}\tilde{F}_\beta(\boldsymbol{q}) &= F_\beta + \sum_{\boldsymbol{s}} p_{\boldsymbol{q}}(\boldsymbol{s}) \log \frac{p_{\boldsymbol{q}}(\boldsymbol{s})}{\mu_\beta(\boldsymbol{s})} \\ &= F_\beta + D_{\mathrm{KL}}(p_{\boldsymbol{q}} \| \mu_\beta) \geq F_\beta,\end{aligned}$$

where $D_{\mathrm{KL}}(\cdot\|\cdot)$ is the Kullback-Leibler divergence which is always greater or equal to zero [Cov99]. Finding the minimum of $\tilde{F}_\beta(\boldsymbol{q})$ with respect to $\boldsymbol{q}$ is the same as minimizing the Kullback-Leibler divergence, which is notoriously a measure of resemblance of two probability distributions.

$$\arg\min_{\boldsymbol{q}} \tilde{F}_\beta(\boldsymbol{q}) = \arg\min_{\boldsymbol{q}} D_{\mathrm{KL}}(p_{\boldsymbol{q}} \| \mu_\beta).$$

The rationale behind the *Bethe* approximation [Bet35; Pei36] is well explained by the following corollary of Lemma 2.1.

Corollary 2.1 (Factorization of $\mu(\boldsymbol{s})$ on a tree)**.** *Under the same hypothesis of Lemma 2.1, the probability distribution $\mu(\boldsymbol{s})$ can be written as*

$$\mu(\boldsymbol{s}) = \prod_{(ij)\in\mathcal{E}} \mu_{ij}(s_i, s_j) \prod_{i\in\mathcal{V}} [\mu_i(s_i)]^{1-d_i},$$

where d_i is the degree of node i.

The proof of Corollary 2.1 is straightforward from Lemma 2.1. The Bethe approximation hence uses as $p_{\boldsymbol{q}}(\boldsymbol{s})$ a distribution that factorizes as in Corollary 2.1. Once again, this approximation is exact on trees and it is expected to be accurate in sparse graphs. Explicitly, we obtain:

$$p^{\text{Bethe}}_{\hat{\boldsymbol{m}},\hat{\boldsymbol{\chi}}}(\boldsymbol{s}) = \prod_{(ij)\in\mathcal{E}} \frac{1+\hat{m}_i s_i + \hat{m}_j s_j + \hat{\chi}_{ij} s_i s_j}{4} \prod_{i\in\mathcal{V}} \left[\frac{1+\hat{m}_i s_i}{2}\right]^{1-d_i}, \quad (2.16)$$

where[7] $\hat{m}_i = \sum_{\boldsymbol{s}} p^{\text{Bethe}}_{\hat{\boldsymbol{m}},\hat{\boldsymbol{\chi}}}(\boldsymbol{s}) s_i$, while $\hat{\chi}_{ij} = \sum_{\boldsymbol{s}} p^{\text{Bethe}}_{\hat{\boldsymbol{m}},\hat{\boldsymbol{\chi}}}(\boldsymbol{s}) s_i s_j$. from this expression the variational Bethe free energy reads

$$\tilde{F}^{\text{Bethe}}_{\beta}(\hat{\boldsymbol{m}},\hat{\boldsymbol{\chi}}) = -\sum_{(ij)\in\mathcal{E}} \beta J_{ij}\hat{\chi}_{ij} + \sum_{(ij)\in\mathcal{E}} \sum_{s_i s_j} f\left(\frac{1+\hat{m}_i s_i + \hat{m}_j s_j + \hat{\chi}_{ij} s_i s_j}{4}\right) + \sum_{i\in\mathcal{V}} (1-d_i) \sum_{s_i} f\left(\frac{1+\hat{m}_i s_i}{2}\right), \quad (2.17)$$

where $f(x) = x\log(x)$. To find the "stable" configurations of $\hat{\boldsymbol{m}},\hat{\boldsymbol{\chi}}$ it is then necessary to minimize the variational free energy with respect to its arguments. Interestingly, the fixed points of BP are minima of the Bethe free energy [MPV87; WF10; Hes+03] drawing a tight relation between the two methods that one could already guess from the relation between Corollary 2.1 and Lemma 2.1. This relation is well explained using the Bethe free energy to determine the stability condition of the paramagnetic phase, introducing a protagonist of this manuscript: the Bethe-Hessian matrix. To accomplish this task, we must verify under what conditions $\hat{\boldsymbol{m}} = \boldsymbol{0}_n$ is a minimum of the variational free energy, so we first impose the gradient of $\tilde{F}^{\text{Bethe}}_{\beta}(\hat{\boldsymbol{m}},\hat{\boldsymbol{\chi}})$ to be equal to zero in $\hat{\boldsymbol{m}} = \boldsymbol{0}_n$.

$$0 = \nabla_{\hat{\boldsymbol{m}}} \tilde{F}^{\text{Bethe}}_{\beta}(\hat{\boldsymbol{m}},\hat{\boldsymbol{\chi}})\Big|_{\hat{\boldsymbol{m}}=\boldsymbol{0}_n}$$
$$0 = \nabla_{\hat{\boldsymbol{\chi}}} \tilde{F}^{\text{Bethe}}_{\beta}(\hat{\boldsymbol{m}},\hat{\boldsymbol{\chi}})\Big|_{\hat{\boldsymbol{m}}=\boldsymbol{0}_n}.$$

The first condition is always verified, while the second imposes $\hat{\chi}_{ij} = \text{th}(\beta J_{ij})$. To determine whether or not $\hat{\boldsymbol{m}} = \boldsymbol{0}_n$ is a minimum of the free energy, we need to compute the Hessian matrix in $\hat{\boldsymbol{m}} = \boldsymbol{0}_n$ that reads [WF10]

The weighted Bethe-Hessian matrix

$$\left(H_{\beta,J}\right)_{ij} \equiv \frac{\partial^2 \tilde{F}^{\text{Bethe}}_{\beta}(\hat{\boldsymbol{m}},\hat{\boldsymbol{\chi}})}{\partial\hat{m}_i \partial\hat{m}_j}\Big|_{\hat{\boldsymbol{m}}=\boldsymbol{0}_n} = \delta_{ij}\left(1 + \sum_{k\in\partial i} \frac{\text{th}^2(\beta J_{ik})}{1-\text{th}^2(\beta J_{ik})}\right) - \frac{\text{th}(\beta J_{ij})}{1-\text{th}^2(\beta J_{ij})}. \quad (2.18)$$

7 The notation $\hat{\boldsymbol{m}}$ is used to stress the fact that $\boldsymbol{m}$ is the exact value of $\langle\boldsymbol{s}\rangle_{\beta}$, while $\hat{\boldsymbol{m}}$ is the one obtained from the Bethe approximation.

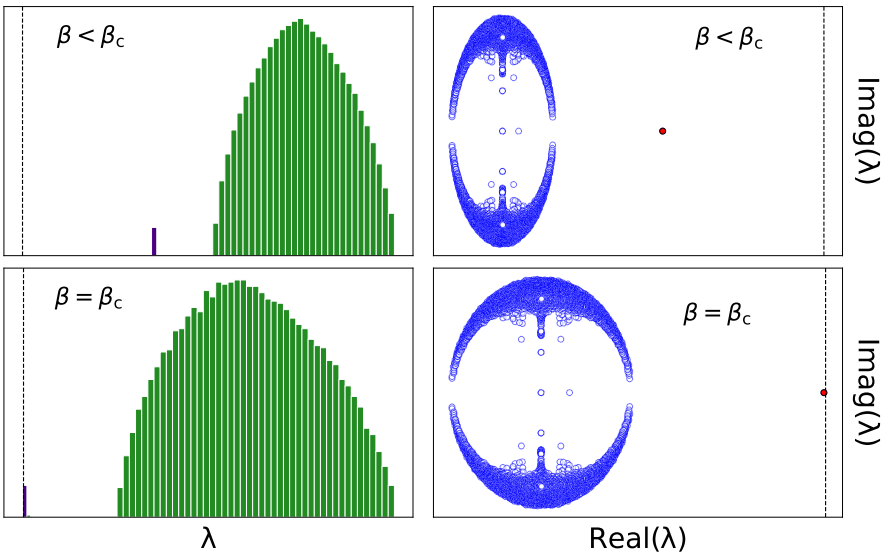


Figure 2.5: **Left**: histogram of the eigenvalues of the matrix $H_{\beta,J}$ (with a zoom on the smallest) for $\beta < \beta_c$ (on top) and $\beta = \beta_c$ on the bottom. The vertical line is at $\lambda = 0$. **Right**: scatter plot of the real and imaginary part of the eigenvalues of $BU_{\beta,J}$, highlighting in red the largest for $\beta < \beta_c$ (on top) and $\beta = \beta_c$ on the bottom. The vertical line is at $\mathrm{Re}(\lambda) = 1$. For this simulation, $J = A$, where A is the adjacency matrix of an ER random graph with $n = 3\,500$ and $c = 5$.

The presence of negative eigenvalues in the spectrum of $H_{\beta,J}$ implies that $\hat{\boldsymbol{m}} = \mathbf{0}_n$ is not the global minimum, hence that the paramagnetic phase is not the stable configuration. Note that, for β sufficiently small, $\hat{\boldsymbol{m}} = \mathbf{0}_n$ is indeed a minimum of the Bethe free energy: in fact $H_{\beta=0,J} = I_n$. In order to determine the condition under which $H_{\beta,J}$ displays negative eigenvalues, it is useful to introduce the following definition of the *generalized Bethe-Hessian* of which $H_{\beta,J}$ constitutes a particular case.

Definition 2.2 (Generalized Bethe-Hessian matrix)**.** *Given a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, a vector $\boldsymbol{\omega} \in \mathbb{R}^{|\mathcal{E}_d|}$ so that $\omega_e = \omega_{e^{-1}}$ and a parameter $r \in \mathbb{C} \setminus \{\pm\omega_e\}_{e\in\mathcal{E}_d}$, the generalized Bethe-Hessian matrix $H_{\boldsymbol{\omega}}(r) \in \mathbb{C}^{n\times n}$ is defined as*

$$(H_{\boldsymbol{\omega}}(r))_{ij} = \delta_{ij}\left(1 + \sum_{k\in\partial i}\frac{\omega_{ik}^2}{r^2-\omega_{ik}^2}\right) - \frac{r\omega_{ij}}{r^2-\omega_{ij}^2}.$$

The matrix $H_{\beta,J}$ then simply corresponds to the choice $r = 1$ and $\omega_{ij} = \mathrm{th}(\beta J_{ij})$. The matrix $H_{\boldsymbol{\omega}}(r)$ has an important relation with the weighted non-backtracking matrix introduced in Section 2.3.1, as stated by the following theorem [WF11; SMM14]:

Theorem 2.1 (Watanabe-Fukumizu formula)**.** *Let B be the non-backtracking matrix of a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ and let $\omega_e \in \mathbb{R}$ be a weight associated to each edge in $\mathcal{E}_d$, with $\omega_e = \omega_{e^{-1}}$. Denote with $U_{\boldsymbol{\omega}} \in R^{|\mathcal{E}_d|}$ the diagonal matrix with entries $U_{\boldsymbol{\omega},ee'} = \delta_{e,e'}\omega_e$. Letting $H_{\boldsymbol{\omega}}(r)$ be the generalized Bethe-Hessian matrix of Definition 2.2, the following relation holds for all $r \in \mathbb{C} \setminus \{\pm\omega_{ij}\}_{(ij)\in\mathcal{E}_d}$*

$$\det\left[rI_{|\mathcal{E}_d|} - BU_{\boldsymbol{\omega}}\right] = \det\left[H_{\boldsymbol{\omega}}(r)\right] \prod_{(ij)\in\mathcal{E}} \left(r^2 - \omega_{ij}^2\right).$$

This theorem is a generalized form of the Ihara-Bass formula [Ter10] which is obtained for $\omega_{ij} = 1$ for all (ij). Exploiting Theorem 2.1, then one obtains the following relation

$$\lambda_1^{\downarrow|\cdot|}(BU_{\beta_c,J}) = 1 \Longrightarrow \det[H_{\beta_c,J}] = 0,$$

where we recall that β_c is the prediction of transition temperature according to BP. The two approximations hence give the same prediction of the transition temperature, since β_c is the smallest β for which $\det[H_{\beta,J}] = 0$. This behavior is schematized in Figure 2.5 in which the spectra of the matrices B and $H_{\beta,J}$ are shown for two different values of β.

To conclude, in this chapter we introduced the basics of statistical physics and of Ising-like models, detailing two methods of fundamental importance for the analysis of the Ising model on sparse graphs. The study of the stability of the paramagnetic phase allowed us to naturally introduce two deeply related matrices that are the non-backtracking and the Bethe-Hessian. We showed that the spectral properties of these matrices are fundamental to draw conclusion on the transition temperatures and a more detailed study of these properties is the object of the next chapter.

3

SPECTRAL PROPERTIES OF THE NON-BACKTRACKING AND BETHE-HESSIAN MATRICES

Abstract

This chapter summarizes some of the most relevant results in random matrix theory *(RMT) to describe the spectral properties of the non-backtracking and Bethe-Hessian matrices of graphs generated from the* degree corrected stochastic block model *(DCSBM). Beyond the rigorous existing results, this chapter further details some heuristic arguments and non-rigorous techniques that have been used in the past (as well as in the remainder of the manuscript) to study the spectrum of the aforementioned matrices.*

In this chapter the spectral properties (*i.e.* the behavior concerning the set of eigenvalues and eigenvectors) of the non-backtracking and Bethe-Hessian matrices introduced in the previous chapters are studied, with a particular attention to the case of large random graphs generated from the *degree corrected stochastic block model* (DCSBM). The concepts and results introduced in this chapter form a bridge between Chapter 2 in which these matrices were given a physical interpretation and the subsequent chapters of Part II and III in which they will be exploited to analyze and propose algorithms for *spectral clustering* (SC). The objective of this chapter is hence twofold: on one side it is to give the rationale behind SC from a *random matrix theory* (RMT) perspective; on the other side it is to introduce the main existing results from the literature of RMT that will be exploited in the remainder.

3.1 BASIC DEFINITIONS

Let us first of all lay out some definitions concerning matrix eigenvalues with a particular focus to large random matrices. In the following, $\lambda_i(M)$ denotes an arbitrary eigenvalue of M. Hermitian matrices have only real eigenvalues that, consequently, can be ordered [Bha13]. We denote in particular with $\lambda_i^{\uparrow}(M)$ the i-th smallest eigenvalue of M and with $\lambda_i^{\downarrow}(M)$ the i-th largest. For non-Hermitian matrices, eigenvalues may have a non-zero imaginary part and $\lambda_i^{\uparrow/\downarrow|\cdot|}(M)$ denotes the i-th smallest/largest eigenvalue *in modulus*. We now define the *empirical spectral measure* $\tilde{\mu}_M(\lambda)$ of a matrix [CD11].

Definition 3.1 (Empirical spectral distribution)**.** *Given a square matrix $M \in \mathbb{C}^{n\times n}$ the empirical spectral measure of its eigenvalues is defined as*

$$\tilde{\mu}_M(\lambda) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}_{\lambda=\lambda_i(M)},$$

where we recall that $\mathbb{I}_x$ is the indicator function. In the particular case in which M is a Hermitian matrix, hence $\lambda_i(M) \in \mathbb{R}$ for all i, $\tilde{\mu}_M(\lambda)$ is the histogram of the eigenvalues of M. The definition of the empirical spectral measure allows us to introduce the fundamental concept of *isolated* and *bulk* eigenvalues on large random matrices.

Definition 3.2 (Bulk and isolated eigenvalues)**.** *Let $M \in \mathbb{C}^{n\times n}$ be a random matrix so that, for all large n with high probability, the empirical spectral measure of M converges to a limit measure,* i.e.

$$\tilde{\mu}_M(\lambda) \underset{n\to\infty}{\longrightarrow} \mu(\lambda).$$

Denoting with $\mathrm{supp}(\mu)$ *the support of the limiting measure and with* $\lambda_{\max} = \max_{\lambda\in\mathrm{supp}(\mu)} |\lambda|$*, all eigenvalues $\lambda_i(M)$ that satisfy*

$$\min_{\lambda\in\mathrm{supp}(\mu)} \left|\frac{\lambda - \lambda_i(M)}{\lambda_{\max}}\right| = o_n(1)$$

are said to be bulk *eigenvalues, while all others are* isolated *eigenvalues.*

SC is based on the existence of isolated eigenvalues

Looking at Definition 3.1, a non vanishing value of $\tilde{\mu}_M(\lambda)$ implies, in the large n limit, that many eigenvalues of M are almost degenerate. When an eigenvalue of M is isolated instead, it is found at a macroscopic distance from all others and it does not contribute to $\tilde{\mu}_M(\lambda)$. This definition is of crucial importance for the remainder, because SC on large random matrices relies on the exploitation of the isolated eigenvalues of a properly chosen matrix. In fact, in simple words, the eigenvectors associated to the isolated eigenvalues of M are related to the expectation of M, while the bulk eigenvalues to its noisy part, as it will be shown in the remainder of this chapter in which this statement will be made more rigorous. Figure 3.1 shows a toy representation of the bulk and isolated eigenvalues of a matrix, together with the corresponding empirical and limiting spectral measures.

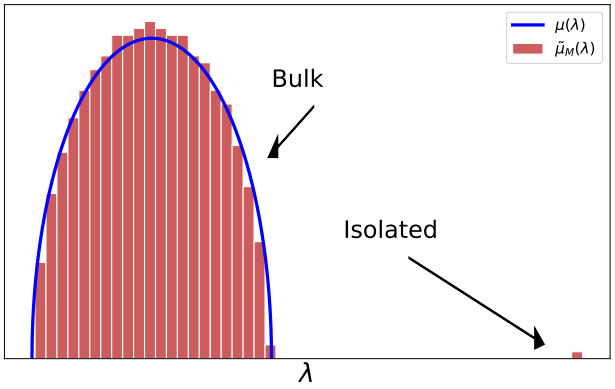


Figure 3.1: Spectrum of the adjacency matrix of a ER random graph with $n = 800$ and $c = 50$, averaged over 10 realizations showing the empirical (in red) and limiting (in blue) distributions.

We now get to the detailed study of some spectral properties, relevant to the results introduced in the next chapters, of the non-backtracking and Bethe-Hessian matrices.

3.2 THE NON-BACKTRACKING MATRIX

First of all, let us recall the definition of the non-backtracking matrix B on a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$. The matrix $B \in \mathbb{R}^{|\mathcal{E}_d| \times |\mathcal{E}_d|}$ is defined on the set of directed edges of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and reads

$$\forall\, (ij), (kl) \in \mathcal{E}_d, \quad B_{(ij),(kl)} = \delta_{jk}(1 - \delta_{il}). \tag{3.1}$$

The non-backtracking matrix hence is non-symmetric also on undirected graphs and consequently its eigenvalues are not necessarily real. In Chapter 2 we also introduced the weighted version of the non-backtracking matrix, obtained from the linearization of *belief propagation* (BP) equations, that can be written as BU_ω, where $U_\omega = \mathrm{diag}(\boldsymbol{\omega}) \in \mathbb{R}^{|\mathcal{E}_d| \times |\mathcal{E}_d|}$ and $\boldsymbol{\omega} \in \mathbb{R}^{|\mathcal{E}_d|}$ is a vector encoding the weight of each directed edge of $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

We first initiate our introduction of the spectral properties of the non-backtracking matrix, limiting ourselves to the unweighted case, to then make further considerations to the more general weighted case.

3.2.1 SPARSE DCSBM

The adoption of the non-backtracking for *spectral clustering* (SC) was proposed in [Krz+13] in which the authors pioneered a method to conjecture some of the spectral properties of B on large sparse *stochastic block model* (SBM) (Definition 1.10) random graphs with $k = 2$ communities of equal size.

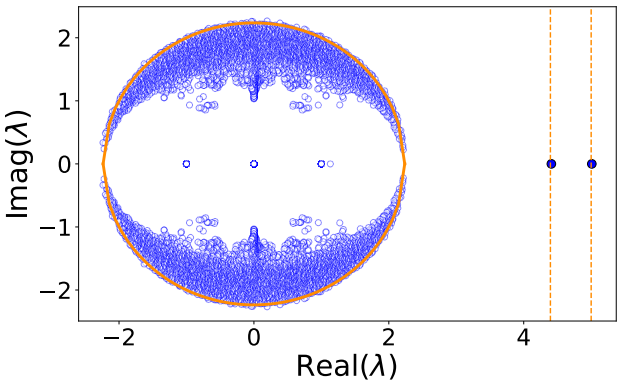


Figure 3.2: Scatter plot in blue of the real vs. imaginary part of the eigenvalues of B of Equation (3.1). For this simulation, $n = 3\,500$, $k = 2$, $c = 5$, $\boldsymbol{\theta} = \mathbf{1}_n$, $c_{\text{out}} = 0.5$. The orange lines are the theoretical prediction of Claim 3.1.

We report here the claim formulated in [Krz+13], together with the heuristic method to derive it that will also be adopted in the subsequent chapters.

Claim 3.1 (Spectrum of B on 2-class sparse SBM)**.** *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a realization of a 2-class SBM as in Definition 1.10 with $\pi_1 = \pi_2$ (the two classes have asymptotically equal size) and $C_{\ell_i,\ell_j} = c_{\text{in}}$ if $\ell_i = \ell_j$ and equals c_{out} otherwise. Denote with $c = (c_{\text{in}} + c_{\text{out}})/2 = O_n(1)$ the expected average degree and suppose $(c_{\text{in}} - c_{\text{out}})/2 > \sqrt{c}$. Then, for all large n, with high probability, the two largest eigenvalues of B in modulus are real, isolated and satisfy*

$$\lambda_1^{\downarrow|\cdot|}(B) = c + o_n(1)$$
$$\lambda_2^{\downarrow|\cdot|}(B) = \frac{c_{\text{in}} - c_{\text{out}}}{2} + o_n(1).$$

All other eigenvalues satisfy $|\lambda_{i\geq 3}^{\downarrow|\cdot|}(B)| \leq \sqrt{c} + o_n(1)$.

The assumption $(c_{\text{in}} - c_{\text{out}})/2 > \sqrt{c}$ is technical and is needed to ensure that $\lambda_2^{\downarrow|\cdot|}(B)$ is isolated. We will come back to this hypothesis in Chapter 4, explaining its deep meaning and relation to information theory. For the moment, however, we treat it as a technical requirement needed to obtain two isolated eigenvalues in the spectrum of B (see Figure 3.2) and proceed detailing the intuitions leading to the conjecture of [Krz+13].

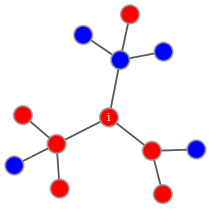
A realization of a sparse SBM locally converges to a tree for which random variables defined on the nodes of the graph are conditionally independent [GLM15]. To exploit this property, we express the probability distribution of the label of a neighbour of an arbitrarily chosen node i, conditionally to ℓ_i using Bayes theorem ,

Bayes theorem $\mathbb{P}(A|B) = \frac{\mathbb{P}(B|A)\mathbb{P}(B)}{\mathbb{P}(A)}$

$$\mathbb{P}(\ell_j|\ell_i, A_{ij} = 1) = \frac{\mathbb{P}(A_{ij} = 1|\ell_i, \ell_j)}{\sum_{\ell_j} \mathbb{P}(A_{ij} = 1|\ell_i, \ell_j)} = \frac{C_{\ell_i,\ell_j}}{2c}. \tag{3.2}$$

So, for each node j in the neighbourhood of i, $\ell_i = \ell_j$ with probability $c_{\text{in}}/(c_{\text{in}} + c_{\text{out}})$ and $\ell_i \neq \ell_j$ with probability $c_{\text{out}}/(c_{\text{in}} + c_{\text{out}})$. Moreover, the variables ℓ_j for $j \in \partial i$ are asymptotically independent conditionally to ℓ_i due to the locally tree-like structure of $\mathcal{G}(\mathcal{V}, \mathcal{E})$. These facts are at the base of Claim 3.1 whose validity is argued in tree steps: (i) define a *guess eigenvector* of B; (ii) show that in expectation (over Equation (3.2)) it is an eigenvector of B; (iii) determine under what conditions the variance of the guess eigenvector vanishes, hence guaranteeing the goodness of the approximation of guess eigenvector with its expectation.

A node in class a has $C_{ab}/2$ neighbours in class b on average



First of all, the guess eigenvector $\boldsymbol{g}^{(r)}$ of B is defined , with r an integer guaranteeing that, for any node $i \in \mathcal{V}$, the sub-graph formed by all nodes that are at most r steps away from i is tree-like.

The guess eigenvector

$$g_{ij}^{(r)} = \frac{1}{\lambda^r} \sum_{\substack{(\omega x)\,:\, d(jk,\omega x)=r \\ k\neq i}} \sigma_x, \tag{3.3}$$

where $\{(jk) \,:\, d(jk, \omega x) = r\}$ is the set of directed edges (jk) such that the shortest non-backtracking path connecting (jk) to (wx) is of length r and $\sigma_x = 1$ if $\ell_x = 1$ and $\sigma_x = -1$ if $\ell_x = 2$. The sum in Equation (3.3) runs over an exponentially large (in r) number of terms hence, for r sufficiently large, one expects good concentration properties of $\boldsymbol{g}^{(r)}$. From a simple calculation, the following identity is verified

"Almost" eigenvalue equation

$$B\boldsymbol{g}^{(r)} = \lambda \boldsymbol{g}^{(r+1)}, \tag{3.4}$$

which means that $\boldsymbol{g}^{(r)}$ is "almost" an eigenvector of B, provided that the dependence on r is small. Furthermore, λ represents the eigenvalue of this approximate eigenvector equation. Studying the expectation of $\boldsymbol{g}^{(r)}$, we determine the value of λ so that $\boldsymbol{g}^{(r)}$ depends on r only marginally. For convenience, $\boldsymbol{g}^{(r)}$ is rewritten in the following form:

$$g_{ij}^{(r)} = \frac{1}{\lambda^r} \sum_{\substack{(q\omega)\,:\, d(jk,q\omega)=r-1 \\ k\neq i}} \sum_{x\in\partial\omega\setminus q} \sigma_x.$$

We now take the expectation of the inner term conditionally to σ_ω:

$$\mathbb{E}\left[\sum_{x\in\partial\omega\setminus q} \sigma_x\right] = \frac{c_{\text{in}} - c_{\text{out}}}{2}\,\sigma_\omega.$$

This gives a recursive formula for $\mathbb{E}[g_{ij}^{(r)}]$, in fact

The expectation of the guess eigenvector

$$\begin{aligned}\mathbb{E}[g_{ij}^{(r)}] &= \frac{1}{\lambda^r} \sum_{\substack{(q\omega)\,:\, d(jk,q\omega)=r-1 \\ k\neq i}} \frac{c_{\text{in}} - c_{\text{out}}}{2}\,\sigma_\omega \\ &= \frac{c_{\text{in}} - c_{\text{out}}}{2\lambda}\,\mathbb{E}[g_{ij}^{(r-1)}] = \ldots = \left(\frac{c_{\text{in}} - c_{\text{out}}}{2\lambda}\right)^r \sigma_j,\end{aligned}$$

which leads to the natural choice $\lambda = (c_{\text{in}} - c_{\text{out}})/2$ that should correspond to an approximate eigenvalue of B as is Equation (3.4). To determine whether or not $\boldsymbol{g}^{(r)}$ is well approximated by its expectation, its variance has to be studied. Let us first take under analysis the second moment, following a similar procedure to the one adopted in [Mos04] for trees.

The variance of the guess eigenvector

$$\mathbb{E}\left[\left(g_{ij}^{(r)}\right)^2\right] = \frac{1}{\lambda^{2r}} \sum_{\substack{(\omega x)\,:\,d(jk,\omega x)=r \\ k\neq i}} \sum_{\substack{(vy)\,:\,d(jk,vy)=r \\ k\neq i}} \mathbb{E}[\sigma_x\sigma_y]$$

$$= \frac{1}{\lambda^{2r}} \sum_{\substack{(\omega x)\,:\,d(jk,\omega x)=r \\ k\neq i}} \left(1 + \sum_{\substack{(vy)\,:\,d(jk,vy)=r \\ k\neq i,(vy)\neq(\omega x)}} \mathbb{E}[\sigma_x\sigma_y]\right).$$

Taking the second term under analysis:

$$\sum_{\substack{(vy)\,:\,d(jk,vy)=r \\ k\neq i,(vy)\neq(\omega x)}} \mathbb{E}[\sigma_x\sigma_y] \approx \sum_{\substack{(vy)\,:\,d(jk,vy)=r \\ k\neq i,(vy)\neq(\omega x)}} \mathbb{E}[\sigma_x]\mathbb{E}[\sigma_y]$$

$$\approx \sum_{\substack{(vy)\,:\,d(jk,vy)=r \\ k\neq i}} \mathbb{E}[\sigma_x]\mathbb{E}[\sigma_y],$$

where we exploited that for $x \neq y$ σ_x is asymptotically independent of σ_y and that the number of paths of length r from (jk) is exponentially large in r: for r sufficiently large (say $r \sim \log(n)$) removing the path $(jk) \to (\omega x)$ is only a minor approximation. One then obtains

$$\mathbb{V}\left[g_{ji}^{(r)}\right] \approx \frac{1}{\lambda^{2r}} \sum_{(\omega x)\,:\,d(jk,\omega x)=r} 1 = \left(\frac{c}{\lambda^2}\right)^r,$$

hence the variance vanishes whenever $\lambda = (c_{\text{in}} - c_{\text{out}})/2 > \sqrt{c}$. Comparing this result with Claim 3.1, this is precisely the hypothesis that guarantees $\lambda_2^{\downarrow|\cdot|}(B)$ to be an isolated eigenvalue of the non-backtracking matrix. Repeating the same procedure, but letting $\sigma_i = 1$ for all i, one gets that $\lambda_i^{\downarrow|\cdot|}(B) = c + o_n(1)$ as stated in Claim 3.1. This short proof shows that the eigenvalue $\lambda_2^{\downarrow|\cdot|}(B)$ is associated to an eigenvector that depends on the class structure of $\mathcal{G}(\mathcal{V},\mathcal{E})$ so long that it is an isolated eigenvalue. When $\lambda_2^{\downarrow|\cdot|}(B) \leq \sqrt{c}$, instead, this approximation is no longer valid and this eigenvalue is lost among the bulk eigenvalues of B.

The claim of [Krz+13] attracted a lot of attention from the mathematical community and in [BLM15] Claim 3.1 was formally proved and extended to the case of $k \geq 2$ communities of arbitrary size. Although formally more rigorous, the idea underlying the proof of [BLM15] follows the steps described above, based on the local convergence of $\mathcal{G}(\mathcal{V},\mathcal{E})$ to a tree. To enunciate the main theorem of [BLM15] let us introduce the following assumption that is formulated under the more general DCSBM hypothesis.

Assumption 3.1. *Let $C \in \mathbb{R}^{k\times k}$ be the class affinity matrix and $\boldsymbol{\pi}$ be the expected class size vector as in Definition 1.11, with $\Pi = \operatorname{diag}(\boldsymbol{\pi})$. Further let $\boldsymbol{\theta}$ be the vector encoding the degree heterogeneity in the DCSBM and denote $\Phi = \mathbb{E}[\theta_i^2]$. The matrices C, Π satisfy the following hypothesis:*

- $C\Pi\mathbf{1}_k = c\mathbf{1}_k$, *for some positive* c,
- $\lambda_p(C\Pi)\Phi > \sqrt{c\Phi}$ *for all* $1 \le p \le k$.

The eigenvalues of $C\Pi$ are real, in fact, for $1 \le p \le k$

$$C\Pi\boldsymbol{v}_p = \lambda_p\boldsymbol{v}$$
$$C\Pi \underbrace{C^{1/2}\boldsymbol{\omega}_p}_{\boldsymbol{v}_p} = \lambda_p \underbrace{C^{1/2}\boldsymbol{\omega}_p}_{\boldsymbol{v}_p}$$
$$\underbrace{C^{1/2}\Pi C^{1/2}}_{\text{Hermitian}}\boldsymbol{\omega}_p = \underbrace{\lambda_p}_{\in\mathbb{R}}\boldsymbol{\omega}_p$$

Let us get deeper into the details of this assumption.

1. First, recall from Definition 1.11 that all entries of C and $\boldsymbol{\pi}$ are positive, hence all entries of $C\Pi$ are positive as well. This implies that there is a non-null probability of connection between any two classes. As a consequence, one may apply the Perron-Frobenius theorem [JH85, Theorems 8.2.2, 8.2.4] on $C\Pi$: its eigenvalue of largest modulus, denoted with c, is positive, simple and its corresponding eigenvector is the only one with all positive entries. By definition $\lambda_1^{\downarrow|\cdot|}(C\Pi) = c$.
2. $C\Pi\mathbf{1}_k = c\mathbf{1}_k$: this assumption imposes that the Perron-Frobenius eigenvector of $C\Pi$ is the vector of all ones, $\mathbf{1}_k$. From Definition 1.11, one can show that this assumption implies that $\mathbb{E}[d]$, the expected average degree, is equal to c. This importantly ensures that the average degree does not depend on the class; in fact, the expected degree of each node i belonging to class p equals $\theta_i(C\Pi\mathbf{1}_k)_p = \theta_i c$, independently of p.
3. The last condition ensures that exactly k isolated eigenvalues appear in the spectrum of B.

With this assumption at hand, we are in position to formally state the main result of [BLM15]

Theorem 3.1 (Spectrum of B on sparse SBM graphs)**.** *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a graph generated from the SBM ($\boldsymbol{\theta} = \mathbf{1}_n$) and satisfying Assumption 3.1 for $c = O_n(1)$, then the following relations are satisfied with high probability:*

$$\forall\, 1 \le i \le k, \quad \lambda_i^{\downarrow|\cdot|}(B) = \lambda_i^{\downarrow|\cdot|}(C\Pi) + o_n(1),$$
$$\forall\, i \ge k+1, \quad |\lambda_i^{\downarrow|\cdot|}(B)| \le \sqrt{c} + o_n(1).$$

For two communities of equal size

$$\lambda_1^{\downarrow|\cdot|}(C\Pi) = c$$
$$\lambda_2^{\downarrow|\cdot|}(C\Pi) = \frac{c_{\text{in}} - c_{\text{out}}}{2}$$

Both the results of [Krz+13] and [BLM15] only consider the SBM, in which a homogeneous degree distribution is assumed. The extension to the case of DCSBM-generated graphs was considered in [GLM17a] for the case of $k = 2$ communities of equal size.

Theorem 3.2 (Spectrum of B on 2-class sparse DCSBM graphs)**.** *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a graph generated from the DCSBM and satisfying Assumption 3.1 for $k = 2$ and $\boldsymbol{\pi} \propto \mathbf{1}_2$ and $c = O_n(1)$, then the following relations are satisfied:*

$$\forall\, 1 \le i \le k, \quad \lambda_i^{\downarrow|\cdot|}(B) = \lambda_i^{\downarrow|\cdot|}(C\Pi)\Phi + o_n(1),$$
$$\forall\, i \ge k+1, \quad |\lambda_i^{\downarrow|\cdot|}(B)| \le \sqrt{c\Phi} + o_n(1).$$

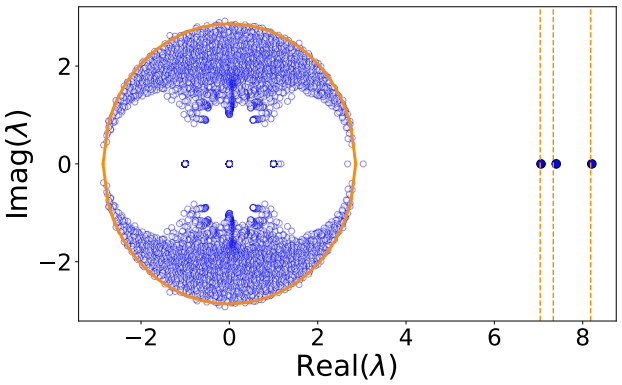


Figure 3.3: Scatter plot in blue of the real vs. imaginary part of the eigenvalues of B of Equation (3.1). For this simulation, $n = 3\,500$, $k = 3$, $c = 5$, $\theta_i \sim [\mathcal{U}(3,10)]^3$. The orange lines are the theoretical prediction corresponding to $\lambda^{\downarrow|\cdot|}_{1\leq i\leq k}(C\Pi)\Phi$ (vertical lines) and $\sqrt{c\Phi}$ (circle).

Note that, Theorem 3.2 holds for two classes of equal sizes, but we believe, motivated by a vast empirical analysis shown[1] in Figure 3.3 and Part II, that Theorem 3.2 has a straightforward generalization in the case only Assumption 3.1 is verified, so for $k \geq 2$ classes of arbitrary size.

With this result we conclude the section on the spectral properties of B on sparse DCSBM graphs and briefly move to the dense setting.

3.2.2 DENSE DCSBM

The main focus of this manuscript is devoted to the sparse regime in which the graph average degree $c = O_n(1)$. As it has been previously discussed, however, the asymptotic concept of sparsity is theory-related and its validity is debatable on real, finite size graphs. Even if all intuitions motivating the non-backtracking matrix as well as the analysis of Section 3.2.1 seem to have as a strict requirement that $\mathcal{G}(\mathcal{V},\mathcal{E})$ has a locally tree-like structure, it appears that the spectral behavior of B on dense DCSBM graphs is very much alike the one obtained in sparse graphs. For this reason, the non-backtracking matrix constitutes a natural bridge between the two regimes as it will be more extensively commented in Chapter 8.

From Theorem 2.1, it appears that all eigenvalues r of B different from ± 1 have to satisfy the following relation on an arbitrary graph $\mathcal{G}(\mathcal{V},\mathcal{E})$.

Quadratic eigenvalue problem

$$\det\left[(r^2-1)I_n + D - rA\right] = 0. \tag{3.5}$$

1 Here and in the following, the notation $\theta_i \sim [\mathcal{U}(a,b)]^\gamma$ indicates the following procedure: (i) n random numbers $\{r_i\}_{i=1,\dots,n}$ are independently drawn from a uniform distribution in a, b; (ii) each of these number are raised to the power γ; (iii) $\theta_i = nr_i / \sum_i r_i$, so that $\mathbb{E}[\theta_i] = 1$.

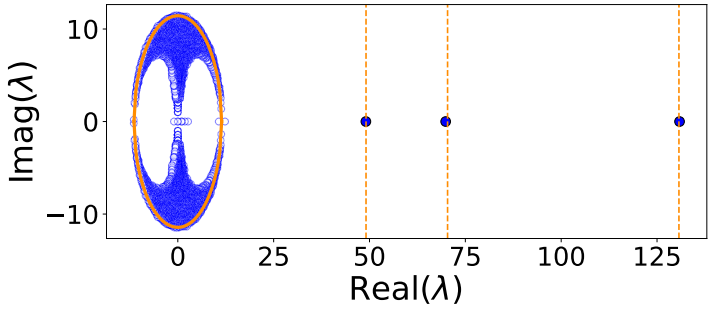


Figure 3.4: Scatter plot in blue of the real vs. imaginary part of the eigenvalues of B of Equation (3.1). For this simulation, $n = 3\,500$, $k = 3$, $c = 80$, $\theta_i \sim [\mathcal{U}(3,10)]^3$. The orange lines are the theoretical prediction corresponding to $\lambda^{\downarrow|\cdot|}_{1\leq i\leq k}(C\Pi)\Phi$ (vertical lines) and $\sqrt{c\Phi}$ (circle).

The solution of Equation (3.5) is known as *quadratic eigenvalue problem* [TM01] and allows one to relate the eigenvalues of B to those of the smaller (unweighted) Bethe-Hessian matrix $H_r = (r^2-1)I_n + D - rA$.

Now, let us first consider a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ generated from the SBM with average degree c going to infinity faster than $\log(n)$. Among the properties of $\mathcal{G}(\mathcal{V},\mathcal{E})$ enumerated in Chapter 1, we recall here that its degree sequence is almost regular, *i.e.* that $\|D - cI_n\| = o_n(c)$. It can be shown [WW17] that the values of r solving Equation (3.5) can be approximated by $\tilde{r}$ solving

$$\det[(\tilde{r}^2 - 1 + c)I_n - \tilde{r}A] = 0. \tag{3.6}$$

To obtain Equation (3.6) from Equation (3.5), D has been replaced by cI_n. The solution of Equation (3.6) is easier to analyze, since it trivially relates $\tilde{r}$ to the eigenvalues of A:

On regular graphs, the eigenvalues of B are directly related to the ones of A

$$\tilde{r} = \frac{\lambda_i(A) \pm \sqrt{\lambda_i^2(A) - 4(c-1)}}{2}. \tag{3.7}$$

Considering the simplest case of two classes of equal size (as in Claim 3.1) the eigenvalues of A satisfy the following identities with high probability (for instance [CZ20, Proposition 3.1])

$$\begin{aligned}
\lambda_1^{\downarrow|\cdot|}(A) &= c + o_n(c)\\
\lambda_2^{\downarrow|\cdot|}(A) &= \frac{c_{\text{in}} - c_{\text{out}}}{2} + o_n(c)\\
|\lambda_{i\geq 2}^{\downarrow|\cdot|}(A)| &\leq 2\sqrt{c} + o_n(\sqrt{c}).
\end{aligned}$$

Combining the spectral properties of A, with Equation (3.7) and the fact that $\tilde{r}$ is a close approximation of r gives the main steps of the informal proof of the following theorem [CZ20].

Theorem 3.3 (Spectrum of B on 2-class dense SBM graphs). *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a graph generated from the SBM and satisfying Assumption 3.1 for $k = 2$, $\boldsymbol{\pi} \propto \mathbf{1}_2$ and $c/\log(n) \to \infty$, then the following relations are satisfied with high probability*

$$\forall\, 1 \leq i \leq k, \quad \lambda_i^{\downarrow|\cdot|}(B) = \lambda_i^{\downarrow|\cdot|}(C\Pi) + o_n(c),$$
$$\forall\, i \geq k+1, \quad |\lambda_i^{\downarrow|\cdot|}(B)| \leq \sqrt{c} + o_n(\sqrt{c}).$$

Once again, the enunciation of Theorem 3.3 uses a notation that can be easily generalized to $k > 2$. Interestingly, however, the empirical evidence suggests that Theorem 3.3 can also be generalized to the DCSBM setting, including $\Phi \neq 1$ as in Theorem 3.2. We give a visual confirmation of this claim in Figure 3.4. With this evidence, let us summarize the two main messages to retain from this section:

- The proofs to obtain the spectral properties of B in the sparse regime heavily exploit its tree-like structure. In the dense regime in which the tree-like hypothesis is not verified, however, the spectral behavior of B is essentially unchanged, hence resilient to the definition of sparsity.
- The proof of Theorem 3.3 requires as a hard constraint the degree concentration that characterizes dense SBM-generated graphs. The empirical evidence however suggests that this is a technical hypothesis and that Theorem 3.3 can be extended also to the DCSBM.

This discussion concludes the properties of sparse DCSBM generated graphs. We now proceed considering the weighted case.

3.2.3 SPARSE WEIGHTED GRAPHS

Recently, in [SM20] the spectrum of B was studied under very generic hypothesis on weighted graphs. Let us recall their main theorem. In the following the notation ' $\circ$ ' is used to indicate the Hadamard entry-wise product.

Theorem 3.4 (Spectrum of BU_ω on weighted random graphs). *Let P be a symmetric random matrix with entries with $0 \leq P_{ij} \leq 1$ and $\max(P_{ij}) = O_n(1/n)$. The adjacency matrix of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is generated setting its entries to one independently (up to symmetry) at random with probability*

$$\mathbb{P}(A_{ij} = 1) = P_{ij}.$$

Let W be a symmetric random matrix with independent entries $W_{ij} = \omega_{ij}$ and denote with $\boldsymbol{\omega} = \{\omega_e\}_{e \in \mathcal{E}_d}$, $U_\omega = \operatorname{diag}(\boldsymbol{\omega})$ and $L = \max(|W_{ij}|)$. Further let $\rho = \lambda_1^{\downarrow|\cdot|}(P \circ \mathbb{E}[W \circ W])$ and r_0 be the number of eigenvalues of $P \circ \mathbb{E}[W]$ that are greater than $\max(\sqrt{\rho}, L)$. The spectrum of the weighted

non-backtracking matrix $B_\omega = BU_\omega$ verifies the following identities with high probability

$$\forall\, 1 \le i \le r_0, \quad \lambda_i^{\downarrow|\cdot|}(B_\omega) = \lambda_i^{\downarrow|\cdot|}(P \circ \mathbb{E}[W]) + o_n(1)$$
$$|\lambda_{i \ge r_0+1}^{\downarrow|\cdot|}(B_\omega)| \le \max(\sqrt{\rho}, L).$$

Although the formulation of Theorem 3.4 is quite involved, the result it conveys is simple: the leading eigenvalues of B_ω are determined by the expectation of the weighted adjacency matrix $P \circ \mathbb{E}[W]$, while the bulk eigenvalues are confined on a circle in the complex plane and are determined by the variance of the weighted adjacency matrix, $P \circ \mathbb{E}[W \circ W]$. This theorem can be seen as a generalization of Theorem 3.1 and 3.2 that are obtained for $W = \mathbf{1}\mathbf{1}_n^T$ and $P_{ij} = \theta_i\theta_j \frac{C_{\ell_i,\ell_j}}{n}$.

With this result, we conclude the properties of the non-backtracking matrix that and briefly summarize some of the (few) known facts concerning the closely related Bethe-Hessian matrix.

3.3 THE BETHE-HESSIAN MATRIX

Unlike the non-backtracking matrices, few rigorous results exist on the spectrum of the Bethe-Hessian matrix, most of which exploit its relation with the non-backtracking matrix as is Theorem 2.1. First of all, let us recall the (generalized) Bethe-Hessian matrix of Definition 2.2

$$(H_\omega(r))_{ij} = \delta_{ij}\left(1 + \sum_{k\in\partial i} \frac{\omega_{ik}^2}{r^2 - \omega_{ik}^2}\right) - \frac{r\omega ij}{r^2 - \omega_{ij}^2}$$

This section gives two important claims on the spectrum of $H_\omega(r)$: (i) the existence of isolated eigenvalues for DCSBM-generated graphs under Assumption 3.1 and (ii) the position of the smallest bulk eigenvalue of $H_\omega(r)$ for a particular choice of r.

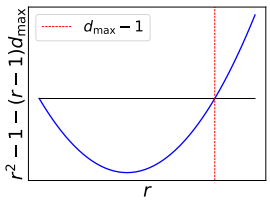
Let us first consider the unweighted case in which $H_\omega(r)$ simply reads

$$H_r = (r^2 - 1)I_n + D - rA.$$

To understand the origin of isolated eigenvalues in the spectrum of H_r, it is enough to note that for a sufficiently large $r \in \mathbb{R}$, thanks to Gershgorin circle theorem [HJ12] $H_r \succ 0$, *i.e.* all its eigenvalues are positive, as already commented along Chapter 2, when studying the stability of the paramagnetic phase of the Ising model. For all $r > \lambda_1^{\downarrow|\cdot|}(B)$, then $H_r \succ 0$ since, by definition $r = \lambda_1^{\downarrow|\cdot|}(B)$ is the largest r for which $\det[H_r] = 0$. At $r = \lambda_1^{\downarrow|\cdot|}(B)$, one eigenvalue of H_r is equal to zero and it is necessarily the smallest one.

The circle theorem implies for $r \in \mathbb{R}$

$$\lambda_1^{\uparrow}(H_r) \ge r^2 - 1 - (r-1)d_{\max}$$



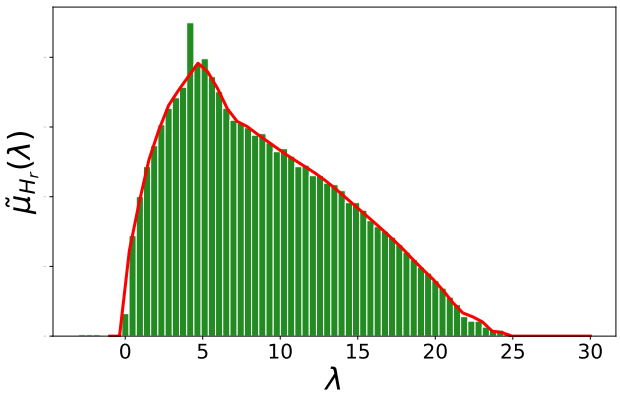


Figure 3.5: Histogram of the eigenvalues of the Bethe-Hessian matrix H_r in green for $r = \sqrt{c}$ on a SBM random graph with expected average degree $c = 5$, $k = 3$ and $n = 5\,000$. In red, the theoretical curve of the bulk eigenvalue limiting distribution as in [SKZ14].

Considering $r = \lambda_1^{\downarrow|\cdot|}(B) \pm \epsilon$ for a small ϵ, it turns out that, again, all eigenvalues of H_r are non-zero, hence the smallest eigenvalue is isolated. Generalizing this reasoning to the following eigenvalues, from an empirical point of view, there is a one-to-one correspondence between the p-th largest (real) eigenvalue of B and the p-th smallest of H_r as stated in the following claim.

The relation between the eigenvalues of H_r and B

Claim 3.2 (Isolated eigenvalues of B and H_r)**.** *Let B and H_r be the non-backtracking and Bethe-Hessian matrices of a realization of a k-class DCSBM satisfying Assumption 3.1. Then, the k leading eigenvalues of B satisfy*

$$1 \leq p \leq k \; : \quad \lambda_p^{\uparrow}\left(H_{\lambda_p^{\downarrow|\cdot|}(B)}\right) = 0.$$

Note that this is not a trivial consequence of Theorem 2.1 which only implies that there exists q such that $\lambda_q^{\uparrow}(H_{\lambda_p^{\downarrow|\cdot|}(B)}) = 0$. This claim instead goes deeper, showing an explicit bijection between the largest (isolated) eigenvalues of B and the smallest (also isolated) of H_r, setting $p = q$.

Claim 3.2 concerns only the isolated eigenvalues of B and H_r. The bulk eigenvalues distribution of the Bethe-Hessian matrix was instead studied in [SKZ14] with the cavity method [Rog+08] for SBM graphs. The result of [SKZ14] allows one to obtain $\mu(\lambda)$ solving a fixed point equation. The result is shown in Figure 3.5. The method of [SKZ14] can be easily generalized to the DCSBM setting, leading in particular the following claim.

Claim 3.3 (Limiting spectral distribution of H_r)**.** *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a realization of the DCSBM verifying Assumption 3.1 for $c = O_n(1)$. Let $\rho = \lambda_1^{\downarrow|\cdot|}(B)$, then the following relation holds with high probability:*

$$\lambda_{k+1}^{\uparrow}\left(H_{\sqrt{\rho}}\right) = 0^+,$$

i.e. *, for r equal to the radius of the bulk of B, the bulk of H_r is asymptotically close to zero (but positive).*

The relation between the eigenvalues of B_ω and those of $H_\omega(x)$ (the mapping between isolated eigenvalues and the relation the bulk edges) holds as well in the weighted regime, as evidenced by extensive simulations that will be more deeply discussed in Part III and Claim 3.2, 3.3 can be generalized to this setting. The theoretical results on the spectrum of $H_\omega(r)$ are very limited comparatively to those obtained for B_ω. Nevertheless, Theorem 2.1 together with Claim 3.2 and 3.3 makes it possible to infer some relevant properties of the spectrum of H_ω as a non-rigorous "corollary" to the theorems characterizing the non-backtracking matrix spectrum.

The relation between the eigenvectors of H_r and B

Finally, let us conclude this section on the spectral properties of the Bethe-Hessian matrix, showing that not only its eigenvalues, but also its eigenvectors are related to those of B.

Proposition 3.1 (Eigenvectors of B_ω and $H_\omega(r)$)**.** *Let B_ω be the weighted non-backtracking matrix of a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ and $H_\omega(r)$ the corresponding generalized Bethe-Hessian matrix. Consider an eigenvector $\boldsymbol{g}$ of B_ω with eigenvalue λ so that $\det[H_\omega(\lambda)] = 0$ and $\lambda \neq |\omega_{ij}|$ for all $(ij) \in \mathcal{E}$.*

Let $\boldsymbol{\psi}(\boldsymbol{g}) \in \mathbb{R}^n$ be defined as

$$\psi_i(\boldsymbol{g}) = \sum_{j\in\partial i} g_{ij}\omega_{ij}.$$

Then $H_\lambda \boldsymbol{\psi}(\boldsymbol{g}) = 0$, i.e. *$\boldsymbol{\psi}(\boldsymbol{g})$ is an eigenvector of H_λ with zero eigenvalue.*

The proof of Proposition 3.1 allows one to prove part of Theorem 2.1 but its main purpose is to explicitly show the relation between the eigenvectors of the two matrices.

Proof. From the definition of $\boldsymbol{\psi}(\boldsymbol{g})$ and of B_ω we can rewrite

$$\psi_i(\boldsymbol{g}) = \sum_{k\in\partial i} \omega_{ik}g_{ik} = \sum_{k\in\partial i\setminus j} \omega_{ik}g_{ik} + \omega_{ij}g_{ij} = (B_\omega \boldsymbol{g})_{ji} + \omega_{ij}g_{ij}.$$

Inverting the role of i and j, the following system is obtained

$$\begin{pmatrix} \lambda & \omega_{ij} \\ \omega_{ij} & \lambda \end{pmatrix} \begin{pmatrix} g_{ji} \\ g_{ij} \end{pmatrix} = \begin{pmatrix} \psi_i(\boldsymbol{g}) \\ \psi_j(\boldsymbol{g}) \end{pmatrix}.$$

Solving this system, g_{ij} can be written as a function of $\psi_i(\boldsymbol{g})$ and $\psi_j(\boldsymbol{g})$:

$$g_{ij} = \frac{1}{\lambda^2 - \omega_{ij}^2}\left[\lambda\psi_j(\boldsymbol{g}) - \omega_{ij}\psi_i(\boldsymbol{g})\right].$$

Reinjecting this relation in the definition of $\psi_i(\boldsymbol{g})$ and exploiting the definition of $H_\omega(r)$, one finally concludes the proof. □

The proof of Proposition 3.1 concludes the study of the spectral properties of the Bethe-Hessian matrix. In the next section we relate the results obtained so far with those introduced in Chapter 2 in which the study of the spectral properties of B_ω was motivated by the Bethe approximation of Ising-like models on sparse graphs.

3.4 RELATIONS WITH THE ISING MODEL

Chapter 2 showed that the leading eigenvalue of the non-backtracking matrix plays a crucial role in determining the stability of the paramagnetic phase in the Ising-like models. In fact, letting β_c satisfy

$$\lambda_1^{\downarrow|\cdot|}(BU_{\beta_c,J}) = 1, \tag{3.8}$$

according to the Bethe approximation, the paramagnetic phase is stable only if $\beta < \beta_c$. Based on this result, we list some important facts.

Percolation threshold on sparse DCSBM

In Chapter 1 we defined the problem of percolation on random graphs as the critical condition (on the parameters of the generative model of the graph) needed to ensure, with high probability, the existence of a giant component in $\mathcal{G}(\mathcal{V},\mathcal{E})$. In Theorem 1.1 we proved in particular that on a DCSBM-generated graph with expected degree independent of the class label, the existence of a giant component in $\mathcal{G}(\mathcal{V},\mathcal{E})$ appears if and only if $c\Phi > 1$. Let us detail some consequences of this fact

- Referring to Assumption 3.1, we immediately notice that the condition $\lambda_p(C\Pi) > \sqrt{c\Phi}$ implies $c\Phi > 1$, since $\lambda_1^{\downarrow}(C\Pi) = c$. Consequently, all theorems and conjectures formulated in this chapter hold for graphs with a giant component.
- According to Theorem 3.2, for a DCSBM-generated graph satisfying Assumption 3.1, the leading eigenvalue of the non-backtracking matrix is asymptotically close to $c\Phi$. The presence of an eigenvalue greater than one in the spectrum of B hence indicates the presence of a giant component. The relation between percolation and the spectrum of B was conjectured already in [KNZ14] in which the authors consider an arbitrary given graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ in which each edge is kept with probability p and determine the critical p to observe a giant component.
- The absence of an eigenvalue of the non-backtracking matrix greater than one makes it impossible to define β_c. Consider for simplicity an unweighted graph, for which Equation (3.8) can be rewritten as

$$\lambda_1^{\downarrow|\cdot|}(B)\mathrm{th}(\beta_c J) = 1.$$

Since $\text{th}(\beta_c J) < 1$ and $\lambda_1^{\downarrow|\cdot|}(B) < 1$ by hypothesis, β_c cannot be defined. This implies that no spontaneous magnetization can be observed is there is no giant component in $\mathcal{G}(\mathcal{V}, \mathcal{E})$, as shown in [Leo+02]. Furthermore, if $c\Phi > 1$, the transition temperature reads

$$\beta_c = \frac{1}{J}\text{ath}\left(\frac{1}{c\Phi}\right).$$

Isolated eigenvectors of H_r

Let us now consider a very important relation between the Bethe free energy (Equation (2.17)) and the eigenvectors of the Bethe-Hessian and non-backtracking matrices, referring, once again, to the unweighted case for simplicity. Since B is a non-negative matrix, the eigenvector associated to its largest eigenvalue (the one defining β_c as in Equation (3.8)), has entries with the same sign. Note also that in Section 3.2.1 the guess eigenvector the eigenvalue is chosen with all positive entries. Consequently, exploiting Claim 3.2 and Proposition 3.1 it turns out that all the entries of the eigenvector associated to $\lambda_1^{\uparrow}(H_{\lambda_1^{\downarrow|\cdot|}(B)})$ have the same sign. Recalling the derivation and physical interpretation of the Bethe-Hessian matrix we detailed in Chapter 2, the eigenvectors of H_r are pointing towards the directions in which minima of the Bethe free energy may appear. The spectral properties of B hence not only give an expression to the transition temperature β_c but also predict that the stable phase is the ferromagnetic one, in which all spins are aligned.

This observation is more general and can be extended to all eigenvectors of H_r (as well as to $H_{\beta,J}$) with a negative eigenvalue. This fundamental intuition will be at the core of our proposed algorithms of Part II and III.

Phase transitions in weighted graphs

In Chapter 2 we showed how the spectral properties of B_ω determine the instability of the paramagnetic phase according to the Bethe approximation. Yet, even if a phase transition is predicted, no other conclusion was made on what happens in general when $\beta > \beta_c$, *i.e.* when the system is no longer in the paramagnetic phase. This is because this question depends generally on the matrix J. In the particular case in which $J = JA$ with $J > 0$ only two phases exist: the paramagnetic and the ferromagnetic. If the system is not in one phase, then it is in the other. This is well explained by the fact that all entries of the eigenvector associate with the smallest eigenvalue of H_r are positive, as we said in the previous paragraph. This property is however a direct consequence of the fact that J has only positive entries. What happens in the more general case in which J_{ij} may be negative?

In this case, we should, more precisely, refer to Equation (2.2) as *spin glass* rather than *Ising* Hamiltonian [MPV87]. The study of spin glasses is a vast field to which we have not the ambition to give an introduction here. We

borrow however an example from the *random bond Ising model* to relate this physical problem to the spectrum of $B_{\boldsymbol{\omega}}$.

Let $\mathcal{G}(\mathcal{V},\mathcal{E})$ be a realization of a Erdős-Rényi random graph and define the following Hamiltonian on $\mathcal{G}(\mathcal{V},\mathcal{E})$.

$$\mathcal{H}(\boldsymbol{s}) = -\sum_{(ij)\in\mathcal{E}} J_{ij}s_i s_j,$$

in which for all $(ij)\in\mathcal{E}$, the J_{ij} are independent random variable distributed according to $p\delta(\mathrm{J}-1)+(1-p)\delta(\mathrm{J}+1)$, for $1/2 \leq p \leq 1$. Let $\boldsymbol{\omega} = \{\mathrm{th}(\beta J_e)\}_{e\in\mathcal{E}_\mathrm{d}}$. From Theorem 3.4 we obtain in this case that

$$\lambda_1^{\downarrow|\cdot|}(B_{\boldsymbol{\omega}}) = \max\left(c(2p-1), \sqrt{c}\,\right)\ \mathrm{th}(\beta).$$

The transition temperature determining the instability of the paramagnetic phase hence occurs at

$$\beta_\mathrm{c} = \mathrm{ath}\left(\frac{1}{\max(c(2p-1),\sqrt{c})}\right).$$

According to the value of p we have two different scenarios, dictated by whether or not $c(2p-1) > \sqrt{c}$. This is precisely the condition that determines the appearance of one isolated eigenvalue in the spectrum of $B_{\boldsymbol{\omega}}$, according to Theorem 3.4. Specifically, for large p (the case $p=1$ reduces to the ferromagnetic Ising Hamiltonian considered in the previous paragraph) the largest eigenvalue of $B_{\boldsymbol{\omega}}$ is isolated and the entries of its eigenvector are positive *in expectation* and when $\beta=\beta_\mathrm{c}$ there is a transition from the paramagnetic to the ferromagnetic phase. For $c(2p-1)<\sqrt{c}$ instead, the largest eigenvalue of $B_{\boldsymbol{\omega}}$ belongs to the bulk and the resulting phase is called *spin-glass* in which the global magnetization is null, but local order is observed.

This discussion closes Part I which was dedicated to the presentation of the main technical tools needed to introduce the original contributions of this manuscript. We now proceed with the first part of our contributions, focused on the problem of community detection in sparse graphs with a heterogeneous degree distribution.

COMMUNITY DETECTION

This part is devoted to the presentation of our contributions to spectral clustering for community detection. First, Chapter 4 provides an overview to the basics of community detection, with a particular attention to the problem of inference in graphs generated from the *degree corrected stochastic block model*. Spectral clustering is then reviewed as an efficient class of algorithms to perform community detection. Chapter 5 details our main contribution to the field, which is stated in the form of a claim which naturally leads to an efficient algorithm for community detection. Successively, Chapter 6 draws an explicit relation between the results of Chapter 5 and some works based on the regularized Laplacian matrix, answering some open questions related to this popular method for spectral clustering. The problem of how to efficiently translate our theoretical results into a practical algorithm is thoroughly treated in Chapter 7. Finally, based on our theoretical results, Chapter 8 serves as a conclusion to Part II, with a critical and retrospective look to spectral clustering for community detection.

4

SHORT OVERVIEW OF SPECTRAL CLUSTERING FOR COMMUNITY DETECTION

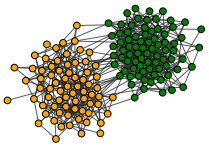
Abstract

This chapter introduces the problem of community detection *(CD) on graphs with a particular attention to the existing contributions approaching CD with* spectral clustering *(SC) algorithms. To this end, the optimization and the inference approaches to CD are compared and the main results concerning inference in the* degree corrected stochastic block model *(DCSBM) are reviewed. In conclusion, a survey on the state-of-the-art SC algorithms for CD is provided, underlining the weaknesses of each of the existing approaches that either are not suited to perform well in the presence of a broad degree distribution or of a low average degree, two properties that typically characterize real-world graphs.*

Part I gave an introduction to graph theory, statistical physics and random matrix theory. These tools are the fundamental building blocks needed to describe our contributions. This chapter is a bridge between Part I and the remainder in which our main results are presented.

In Chapter 1 some properties of real-world graphs have been introduced, recalling, among others, the recurrent presence of a community structure [LM+54; MSLC01] according to which nodes in the same class are more densely connected than nodes in different classes. The problem of *community detection* (CD) can be informally defined as the task of determining a non-overlapping node partition of a graph, unveiling its underlying community structure. This definition is somewhat tautological given its dependence on the concept of *communities* for which there exist no shared consensus in the scientific literature [FH16]. Its clear requirement is that each node has to be assigned to a unique class. More general, and possibly more realistic, definitions in which each node may belong to more than one class exist [XKS13] but will not be considered in the remainder.

A graph with $k = 2$ communities



Some of the earliest approaches [Sco88] to CD relied on the concept of *cliques*, *i.e.* sub-graphs of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ in which all nodes are connected among themselves. This represents the extreme case in which nodes in the same community are maximally connected. Communities, however, can be defined even under less stringent conditions.

Among the several strategies to perform CD [For10; GN02; New04a; FH16] in this chapter we consider two in particular. In the first, the class structure is determined solving an optimization problem. This approach is very popular in the CD literature and inspired some of the earliest works on *spectral clustering* (SC) [VL07]. The second method, instead, considers CD as an inference problem from the *degree corrected stochastic block model* (DCSBM). In Section 4.1 we briefly introduce the key ideas of some of the most commonly deployed definitions of CD as an optimization problem, evidencing, in particular, the major weaknesses of this approach. Subsequently, in Section 4.2, we review the main results concerning inference in the sparse DCSBM and discuss how this approach is capable to overcome the limitations of the optimization strategy. Finally, Section 4.3 serves as a review to state-of-the-art methods of SC for CD. A particular attention will be devoted to how each of the considered algorithm deals (or not) with sparsity, *i.e.* low average degree and heterogeneity in the degree distribution. We recall, from Chapter 1, that these are two very recurrent properties that characterize real graphs and they generally constitute a big challenge for SC.

4.1 COMMUNITY DETECTION AS AN OPTIMIZATION PROBLEM

Defining communities as the solution of an optimization problem consists in identifying a quality function assessing how satisfactory a given class partition is on a graph. Such function should depend on the partition ℓ and the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, in the form of its adjacency matrix A. Some of the earliest choices for the quality function rely on graph cuts, such as the *ratio*

cut (RCut) [HK92], defined in Equation (4.1) and the *normalized cut* (NCut) [SM00], defined in Equation (4.2).

Graph cuts

$$\mathscr{Q}_A^{\text{RCut}}(\boldsymbol{\ell}) = \frac{1}{2}\sum_{p=1}^{k}\frac{1}{|\mathcal{V}_p|}\sum_{i\in\mathcal{V}_p}\sum_{j\notin\mathcal{V}_p}A_{ij} \tag{4.1}$$

$$\mathscr{Q}_A^{\text{NCut}}(\boldsymbol{\ell}) = \frac{1}{2}\sum_{p=1}^{k}\frac{\sum_{i\in\mathcal{V}_p}\sum_{j\notin\mathcal{V}_p}A_{ij}}{\sum_{i\in\mathcal{V}_p}\sum_{j\in\mathcal{V}}A_{ij}}, \tag{4.2}$$

where $\mathcal{V}_p = \{i \in \mathcal{V} : \ell_i = p\}$ is the set of all nodes with label p. According to these cost functions, high values of $\mathscr{Q}_A^{\bullet}$ are obtained when there are several edges connecting nodes in different communities. The goal is hence to find $\boldsymbol{\ell}$ that minimizes $\mathscr{Q}_A^{\bullet}$, under the constraint $\boldsymbol{\ell} \neq \mathbf{1}_n$, which prevents all nodes from being assigned to the same cluster.

Another quality function that is very popular in modern CD is the *modularity*, introduced in [NG04] to evaluate the quality of a community partition:

Modularity

$$\mathscr{Q}_A^{\text{Mod}}(\boldsymbol{\ell}) = \frac{1}{4|\mathcal{E}|}\sum_{i,j\in\mathcal{V}}\left(A_{ij} - \frac{d_i d_j}{2|\mathcal{E}|}\right)\delta(\ell_i,\ell_j), \tag{4.3}$$

where we recall that $\boldsymbol{d} = A\mathbf{1}_n$ indicates the degree vector. The modularity attributes a large score to configurations in which nodes in the same community are connected by a *greater than expected*[1] number of edges. In fact, for a fixed degree sequence, $d_i d_j/2|\mathcal{E}|$ is the probability that nodes i and j are connected if edges were placed at random. The modularity has subsequently been exploited to define CD algorithms in which the label assignment is obtained maximizing the $\mathscr{Q}_A^{\text{Mod}}(\boldsymbol{\ell})$ and it is one of the most widely adopted strategies to CD [Rad+04; New04b; CNM04; Blo+08; TWVE19].

Defining communities according to a score function may seem a particularly good strategy for two main reasons. The first is that the definition of communities does not make any assumption[2] on the matrix A and is hence adapted to any graph. The second is that it gives a common sense definition of communities, expressing a property of a good class assignment. This approach has, however, strong algorithmic limitations.

The pitfalls of the optimization approach to CD

- Optimization problems such as NCut, RCut and the modularity maximization are NP hard[3] and only approximate solutions can be obtained by efficient algorithms [FH16; VL07].

1 In other words, the modularity compares the realization of the matrix A with a typical realization of a *null model*, called *configuration model*.

2 Or better, does not make any *explicit* assumption on the matrix A. In the following we will reconsider this assertion and argue that, matter of factly, the optimization approach makes some implicit assumptions of A.

3 Although this has not been proved but only conjectured, it is believed that NP-hard optimization problems do not admit a polynomial time solution, *i.e.* , for a problem of size n, an exponential in n number of operations is required to obtain the solution [VL91].

- There exists a large number of structurally different configurations having values of $\mathscr{Q}_A^\bullet$ value very close to the maximum [GDMC10]. Running multiple times an approximate algorithm looking for the optimum of $\mathscr{Q}_A^\bullet$ may output similar results in terms of the score, but corresponding to rather different label assignments.

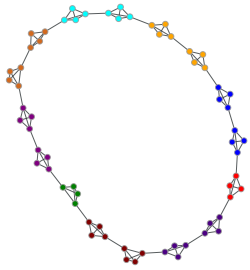
- The authors of [FB07] showed that, even in the presence of well defined clusters (such as *cliques*), optimizing the modularity score over the number of clusters k, small communities may be joined together, causing the so-called *resolution limit*. This is a consequence of the fact that the values of $\mathscr{Q}_A^{\text{Mod}}$ are not directly comparable for different values of k. Albeit the authors of [FB07] only consider the *modularity*, they claim that similar effects are expected to be seen also when optimizing other cost functions, such as $\mathscr{Q}_A^{\text{RCut}}$ or $\mathscr{Q}_A^{\text{NCut}}$. To circumvent this problem, in [AFG08] it was introduced a generalized modularity, depending on a positive regularizer γ:

$$\mathscr{Q}_A^{GMod}(\boldsymbol{\ell};\gamma) = \frac{1}{4|\mathcal{E}|}\sum_{i,j\in\mathcal{V}}\left(A_{ij} - \gamma\frac{d_i d_j}{2|\mathcal{E}|}\right)\delta(\ell_i,\ell_j). \tag{4.4}$$

  Tuning the value γ it is possible to identify communities at different length scales, but this requires an *ad hoc* solution, depending, in general, on the underlying graph.

- From the optimization perspective, communities can be defined even on graphs with no community structure, such as *Erdős-Rényi* (ER) random graphs. This is not only a philosophical problem, related to the fact that a good CD algorithm should be capable of detecting whether or not communities are present on the graph. In fact, considering for instance the modularity, one expects that on ER graphs, any partition satisfies $\mathscr{Q}_A^{\text{Mod}} \approx 0$. It has however been shown in [GSPA04] that high modularity partitions can be found on ER graphs, evidencing that the modularity maximization may lead to over-fitting.

The effect of the resolution limit is well evidenced by ring of cliques *graph: here the colour is the assignment obtained maximizing the modularity*



These problems altogether are severe limitations of the optimization approach to CD and justify the adoption of a different strategy, based on inference on the DCSBM. We will show in the next section how Bayesian inference is able to overcome the aforementioned limitations of optimization and how it is able to motivate their origin.

4.2 INFERENCE IN THE DCSBM

4.2.1 BASIC PROPERTIES

Let us recall that, according to Definition 1.11, the DCSBM generates the adjacency matrix of $\mathcal{G}(\mathcal{V},\mathcal{E})$ setting to one its entries independently (up to symmetry) at random with probability

$$\mathbb{P}(A_{ij} = 1 | \ell_i, \ell_j) = \min\left(\theta_i \theta_j \frac{C_{\ell_i, \ell_j}}{n}, 1\right). \tag{4.5}$$

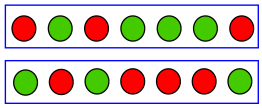
We recall that $C \in \mathbb{R}^{k \times k}$ is the class affinity matrix, generating the community structure of $\mathcal{G}(\mathcal{V}, \mathcal{E})$, while the vector $\boldsymbol{\theta} \in \mathbb{R}^n$ creates an arbitrary degree distribution with the expected degree of note i being proportional to θ_i. Furthermore, the vector $\boldsymbol{\pi} \in \mathbb{R}^k$ is defined so that π_p is the expected fraction of nodes with label p and we denote with $\Pi = \text{diag}(\boldsymbol{\pi})$. We work under Assumption 3.1, according to which $\mathbf{1}_k$ is an eigenvector of $C\Pi$ with eigenvalue c, representing the (class-independent) expected average degree.

Given the generative model of Equation (4.5) the labels are indisputably defined by the vector $\boldsymbol{\ell}$ and the number of classes by k. The goal of CD given a DCSBM-generated graph is hence to infer $\boldsymbol{\ell}$. Let us define the *overlap* to characterize more formally CD on DCSBM graphs.

Definition 4.1 (Overlap). *Let $\boldsymbol{\ell}, \hat{\boldsymbol{\ell}} \in \{1, \dots, k\}^n$ be two vectors and $\mathcal{P}_\ell$ the set of all permutations of $\boldsymbol{\ell}$. The overlap between $\boldsymbol{\ell}$ and $\hat{\boldsymbol{\ell}}$ is defined as*

$$\mathcal{Q}^{\text{ov}}(\boldsymbol{\ell}, \hat{\boldsymbol{\ell}}) = \frac{1}{1 - \frac{1}{k}} \max_{\mathfrak{p} \in \mathcal{P}_\ell} \left[\frac{1}{n} \sum_{i=1}^{n} \left(\delta(\hat{\ell}_i, \mathfrak{p}(\ell_i)) - \frac{1}{k}\right)\right].$$

Two permutations of the same solution

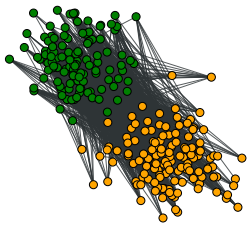


In words, the term $\frac{1}{n}\sum_{i=1}^{n} \delta(\hat{\ell}_i, \mathfrak{p}(\ell_i))$ represents the fraction of correctly labelled nodes; the term $1/k$, instead is the probability that $\delta(\hat{\ell}_i, \mathfrak{p}(\ell_i)) = 1$ if the entries of $\hat{\boldsymbol{\ell}}$ were randomly assigned. For all $\boldsymbol{\ell}$, the function $\mathcal{Q}^{\text{Ov}}(\boldsymbol{\ell}, \hat{\boldsymbol{\ell}})$ is between 0 (random label assignment) and 1 (perfect label assignment). Note that, since we are taking the maximum over all label permutations, $\mathcal{Q}^{\text{ov}}$ can never be negative.

The overlap being defined, CD can be formulated as the task of finding a partition $\hat{\boldsymbol{\ell}}$ so that $\mathcal{Q}^{\text{ov}}(\boldsymbol{\ell}, \hat{\boldsymbol{\ell}}) > 0$ [Abb17]. In words this means to infer $\boldsymbol{\ell}$ better than random guess.

Remark 4.1 (Detection in sparse graphs). *Note that on sparse DCSBM-generated graphs, with high probability, $\min_i(d_i) = 0$,* i.e. *there are isolated nodes that cannot be classified better than random guess [MNS14]. For this reason, when considering the sparse setting, it is most appropriate to talk about detection and not exact recovery that is structurally unfeasible.*

An example of a disassortative *graph*



The class structure of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is simply determined by the matrix C, imposing that $C_{aa} > C_{ab}$ for $b \neq a$. This results in an adjacency matrix in which fewer edges connect nodes in different classes, as required by the intuitive definition of communities. Note however that, even though it is a typical requirement that the diagonal elements of C are greater than the off-diagonal ones, the DCSBM can also be used to describe *disassortative* community structures. In this case groups of nodes are identified as a class because they repel (rather than attract) each other. As a concrete example one may think of the vertices of a graph as the words contained in a text with edges

if two words appear next to each other: on this graph, adjectives and nouns represent two disassortative communities [NG04]. In the case of two classes, this corresponds to imposing $C_{aa} < C_{ab}$. While the definition of a disassortative community structure with the DCSBM is possible, in the remainder we will mainly focus on the assortative case.

Coming to the details of CD as an inference problem, the probability distribution of the labels $\boldsymbol{\ell}$ given the graph observation A can be retrieved with a Bayesian approach. In particular

Bayes' theorem

$$\overbrace{\mathbb{P}(\boldsymbol{\ell}|A)}^{\text{Posterior}} = \frac{\overbrace{\mathbb{P}(A|\boldsymbol{\ell})}^{\text{Likelihood}}\overbrace{\mathbb{P}(\boldsymbol{\ell})}^{\text{Prior}}}{\underbrace{P(A)}_{\text{Evidence}}}.$$

The *prior* $\mathbb{P}(\boldsymbol{\ell})$ encodes pre-existing information on the class label assignment. We suppose nothing is known about the community structure and consequently the uniform $\boldsymbol{\ell}$-independent prior is chosen. The *likelihood* is obtained from Equation 4.5, while the *evidence* is an $\boldsymbol{\ell}$-independent normalization constant. For the DCSBM, assuming $\theta_i\theta_j C_{\ell_i,\ell_j}/n < 1$ for all i, j, the posterior distribution reads:

$$\begin{aligned}\mathbb{P}(\boldsymbol{\ell}|A) &\propto \prod_{(ij)\in\mathcal{E}} \theta_i\theta_j\frac{C_{\ell_i,\ell_j}}{n}\cdot\prod_{(ij)\notin\mathcal{E}}\left(1-\theta_i\theta_j\frac{C_{\ell_i,\ell_j}}{n}\right)\\ &\propto \exp\left\{\sum_{(ij)\in\mathcal{E}}\log\left(C_{\ell_i,\ell_j}\right)+\sum_{(ij)\notin\mathcal{E}}\log\left(1-\theta_i\theta_j\frac{C_{\ell_i,\ell_j}}{n}\right)\right\}.\end{aligned} \tag{4.6}$$

Obtaining the marginal node probability from Equation (4.6), one can assign to each node the label maximizing the node marginal [Pei19]. A possible way to accomplish this task is to sample from the distribution (4.6) using Monte Carlo Markov chains [Pei14a]. The Bayes optimal procedure is, however, typically quite expensive from the computational viewpoint. Luckily, for sparse graphs, the asymptotically exact expression of $\mathbb{P}_i(\ell_i|A)$ can be efficiently obtained using the *belief propagation* (BP) algorithm introduced in Chapter 2. The node marginals are obtained solving the following system of equations[4] [Dec+11, Eq. 26, 27, 28]:

BP for CD

$$\mathbb{P}_i(\ell_i|A) = \frac{\pi_{\ell_i}}{Z_i}e^{-h_{\ell_i}}\prod_{j\in\partial i}\sum_{\ell_j}C_{\ell_i,\ell_j}\eta_{ij}(\ell_j)+o_n(1) \tag{4.7}$$

$$\eta_{ji}(\ell_i) = \frac{1}{Z_{ji}}\pi_{\ell_i}e^{-h_{\ell_i}}\prod_{k\in\partial i\setminus j}\sum_{\ell_k}C_{\ell_k,\ell_i}\eta_{ik}(\ell_k) \tag{4.8}$$

4 Note that the field h_{ℓ_i} comes from the contribution of all the "non edges" appearing in Equation (4.6) and this expression specifically holds in the sparse SBM.

$$h_{\ell_i} = \frac{1}{n} \sum_{k \in \mathcal{V}} \sum_{\ell_k} C_{\ell_i, \ell_k} \mathbb{P}(\ell_k | A). \tag{4.9}$$

Note that Equations (4.7, 4.8, 4.9) are obtained for the *stochastic block model* (SBM) (which we recall corresponds to the choice $\boldsymbol{\theta} = \mathbf{1}_n$) but the result can be easily generalized to the DCSBM. Using BP propagation it is also possible to iteratively learn the parameters C and $\boldsymbol{\pi}$ that cannot be supposed to be known in advance.

Remark 4.2 (The estimation of k)**.** *In [Dec+11] the authors explain how to estimate the number of communities k if not known in advance. The procedure consists in running BP multiple times for different values of k and keeping the one minimizing the Bethe free energy. As shown in [Pei19], however, when the underlying graph is very far from a realization of the SBM – like a ring of cliques – the estimate of k may become inaccurate, suffering from a* resolution limit.

The computational complexity of BP for CD scales as $O(|\mathcal{E}|k^2)$ [Pei19], making BP computationally efficient in the sparse regime in which $c = O_n(1)$ (or, equivalently, $|\mathcal{E}| = O_n(n)$). The main interest in BP, however, comes from the fact that it is asymptomatically exact in the sparse regime. Studying the convergence of BP allowed the authors of [Dec+11] to formulate some deep conjectures on the theoretical feasibility of inference in the sparse SBM, claiming that if BP is not capable of obtaining a non-zero overlap partition, then no other efficient algorithm can.

4.2.2 DETECTABILITY THRESHOLD

One of the most striking results of [Dec+11] concerns the conjecture of the existence of a *detectability threshold*[5] when inferring the labels. Consider the case of k communities of equal size with $C = c_{\text{out}} \mathbf{1}_k \mathbf{1}_k^T + (c_{\text{in}} - c_{\text{out}}) I_k$. The expected average degree is $c = (c_{\text{in}} + c_{\text{out}}(k-1))/k$. We define α as:

$$\alpha = \frac{c - c_{\text{out}}}{\sqrt{c}}. \tag{4.10}$$

Algorithmic and information theoretic thresholds

The authors of [Dec+11] conjecture that for $\alpha > \alpha_c = 1$ there exist efficient algorithms that are capable to perform CD, obtaining a non-zero overlap. In particular, $\alpha > \alpha_c$ is the condition needed by BP to converge towards an informative fixed point. What happens when $\alpha < \alpha_c$ is slightly more involved. In particular, for any k, BP is not (asymptotically) capable of achieving an overlap greater than zero if a random initialization is chosen. It is supposed that no other efficient algorithm (*i.e.* with computational complexity polynomial in n) can achieve this task. Hence, $\alpha = \alpha_c$ is the *algorithmic threshold* of CD for the SBM. Yet, under proper conditions, it is observed that

5 Note that the existence of a detectability threshold had been previously conjectured in [RL08] in which the authors, however, overestimate the size of the undetectable region.

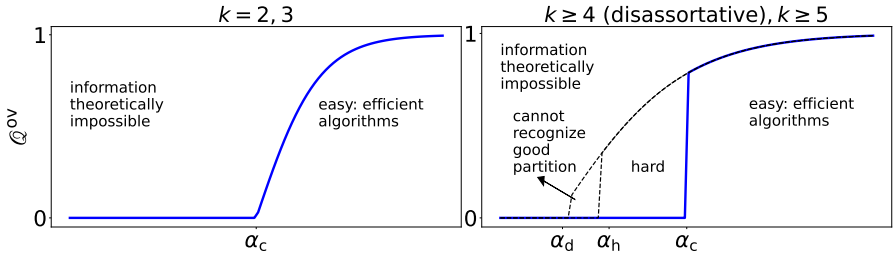


Figure 4.1: Schematic representation of the phase transition in the sparse SBM. Picture adapted from [Moo17, Figure 7].

for an initialization sufficiently close to the informative fixed point, BP obtains a positive overlap even for $\alpha < \alpha_c$. When BP cannot obtain a positive overlap even when initialized to the ground-truth assignment, we are at the *information theoretic threshold* below which there is no way to distinguish a SBM from an *Erdős-Rényi* (ER) random graph.

In the case of $k = 2, 3$ communities it is claimed that the *algorithmic* and *information theoretic* thresholds coincide, hence that for $\alpha < \alpha_c$ no algorithm can (asymptotically) obtain a positive overlap . For $k \geq 4$ on disassortative graphs and $k \geq 5$ on assortative ones, multiple transitions exist:

- $\alpha < \alpha_d$: if BP is initialized to the ground truth assignment, it converges to the factorized fixed point $\mathbb{P}_i(\ell_i) = 1/k$: detection is unfeasible.
- $\alpha_d < \alpha < \alpha_h$: if BP is initialized to the ground truth assignment, it converges to a solution with positive overlap. However, there is no way to distinguish this solution from any other: detection is unfeasible.
- $\alpha_h < \alpha < \alpha_c$: if BP is initialized to the ground truth assignment, it leads to a partition with positive overlap which is distinguishable from all others. Detection is feasible, but it requires an exponential time.

Figure 4.1 summarizes the phase transitions. Since its appearance, the conjectures of [Dec+11] conveyed a lot of attention and several authors contributed to give them a rigorous proof. In particular, [Mas14; MNS18] focused on the SBM for $k = 2$ communities, independently showing that detection can be efficiently obtained when $\alpha > \alpha_c$. In the same setting [MNS15] instead proved the negative part of the conjecture, *i.e.* that detection is impossible when $\alpha < \alpha_c$. Moreover, [MNS14] proved that a modified version of BP maximizes the fraction of nodes labelled correctly for some $\alpha > \alpha_c$. The extension of the possibility part for $k > 2$ was proved in [AS15], while in [CO+18] the conjecture was proved for disassortative graphs. Finally, the results of [Dec+11] have been extended to the $k = 2$ class setting on DCSBM-generated graphs. In this case α should be re-defined as $\alpha = (c - c_{\text{out}})\sqrt{\frac{\Phi}{c}}$, where we recall that $\Phi = \mathbb{E}[\theta_i^2]$. In [GLM15] it was shown that detectability is impossible if $\alpha < \alpha_c = 1$, while the positive part was proved in [GLM17a].

4.2.3 OPTIMIZATION VS MODEL BASED: A STATISTICAL PHYSICS PERSPECTIVE

To conclude this section, let us relate the model-based and optimization-based approaches.

As it was described in Section 4.1, defining communities as the solution to an optimization problem makes a requirement on what a good class partition should be like, with no hypothesis on the underlying graph. This makes it a seemingly good way of performing CD on arbitrary graphs. On the opposite, designing an algorithm for CD inspired from DCSBM gives a good mathematical control and, in some cases, information theoretic guarantees. Nevertheless, the model-based approach relies on some assumptions that are not necessarily verified on arbitrary graphs: recall for instance the locally tree-like structure of sparse DCSBM graphs.

This may lead to thinking of the inference approach as a mere mathematical exercise. This section on the opposite argues that the model-based approach should be generally preferred to the optimization one. In fact, the latter actually relies on some implicit hypothesis on the matrix A and its limitations can be clearly interpreted from a Bayesian perspective.

To simplify the discussion, let us consider the $k = 2$ class DCSBM. In this case, letting $\sigma_i = 1$ if $\ell_i = 1$ and $\sigma_i = -1$ if $\ell_i = 2$, the posterior probability $\mathbb{P}(\boldsymbol{\ell}|A) \equiv \mathbb{P}(\boldsymbol{\sigma}|A)$ of Equation (4.6) can be rewritten as

$$\mathbb{P}(\boldsymbol{\sigma}|A) = \frac{1}{Z} \exp\left\{ \sum_{(ij)\in\mathcal{E}} \beta\sigma_i\sigma_j - \sum_{i\in\mathcal{V}} h_i(\boldsymbol{\sigma})\sigma_i \right\} \equiv \frac{e^{-\beta\mathcal{H}(\sigma)}}{Z}, \qquad (4.11)$$

where $\beta = \frac{1}{2}\log\left(\frac{c_{\text{in}}}{c_{\text{out}}}\right)$ and $h_i(\boldsymbol{\sigma})$ is a local magnetic field due to non-edges that prevents all nodes from ending up in the same class. Equation (4.11) precisely corresponds to the Boltzmann distribution for the Ising Hamiltonian with local fields, depending on the configuration $\boldsymbol{\sigma}$. The Bayesian approach is equivalent to finding the magnetization $\boldsymbol{m} = \langle \boldsymbol{s} \rangle_\beta$, associated to the Hamiltonian $\mathcal{H}(\boldsymbol{\sigma})$. Finding the ground state of $\mathcal{H}(\boldsymbol{\sigma})$, *i.e.* the configuration $\boldsymbol{\sigma}$ corresponding to its minimum, instead is equivalent to finding the maximum of the generalized modularity $\mathscr{Q}_A^{GMod}(\boldsymbol{\ell};\gamma)$, for a proper value of γ, as shown in [New16].

This observation puts us in position to make two very important remarks. The most questionable assumption of the DCSBM is that of generating edges independently at random. In terms of log-likelihood, this translates into a sum over all graph edges, as shown in Equation (4.11). This sum is the same appearing in $\mathscr{Q}_A^{\text{GMod}}$, $\mathscr{Q}_A^{\text{Mod}}$, $\mathscr{Q}_A^{\text{RCut}}$ and $\mathscr{Q}_A^{\text{NCut}}$ that can be associated to a generative model (different from the DCSBM) in which the edges of $\mathcal{G}(\mathcal{V},\mathcal{E})$ are also generated independently at random, evidencing how the functions $\mathscr{Q}_A^{\bullet}$ rely on some "silent" assumptions on the matrix A.

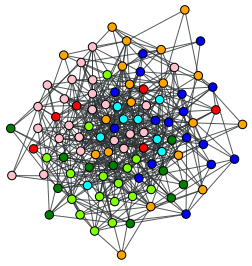
Furthermore, from a statistical physics perspective, the functions $\mathscr{Q}_A^\bullet$ (eventually taken with a negative sign) can generally be considered as Hamiltonians, *i.e.* cost functions associated to a given label configuration. In all cases (also for the DCSBM), the Hamiltonian is what *defines* communities from a *microscopic* perspective. Stating which definition is the best is a difficult task that we are not going to investigate. However it should be remarked that what is *essentially* different in the optimization and inference approaches is how to retrieve the communities from the Hamiltonian: in one case they are obtained from the marginals of the Boltzmann distribution, in the other from the ground state energy.

At this point the natural question to ask is "what is best?". Finding a definitive answer to this puzzle is beyond the scope of this manuscript. The answer is likely to be problem-dependent [Pei19] and, although we do not claim to provide universal arguments, we give two reasons why, at least in our setting, the Bayesian approach should be preferred.

- The Hamiltonian, or equivalently the cost $\mathscr{Q}_A^\bullet$ is what *defines* the concept of communities. The optimization approach, however, only takes into account the minimum of the Hamiltonian, disregarding the rest of its profile. Consider two functions $\mathscr{Q}_A^\bullet$, one being convex and the other having multiple minima with similar values of $\mathscr{Q}_A^\bullet(\boldsymbol{\ell})$. These two settings are clearly different: in the first the label assignment is uniquely defined by minimum of $\mathscr{Q}_A^\bullet$, whereas, in the second, several configurations could be considered as almost equally good community structures. Taking only the minimum of the Hamiltonian means to disregard all other configurations that may have, instead, a potentially great importance. The Bayesian approach does not consider exclusively the minimum of $\mathscr{Q}_A^\bullet$, but the whole energy landscape, giving a generally richer description of the problem.

- The solution of the optimization problem does not involve the confidence that one can have in the solution. To get more practical, consider a set of realizations of the DCSBM with the same ground-truth labels. Running an optimization algorithm on each of these graphs independently will likely lead to different partitions, especially as the value of α decreases.[6] Using the Bayes rule to determine the node marginals, instead, gives in expectation very similar results of the marginals, making this method resilient to noise. This can be well understood by taking a closer look at Equation (4.11). For hard detection problems, $\beta \to 0$, leading the system into the *paramagnetic* phase in which no clustering can be made. The Bayes approach hence correctly predicts the absence of communities in *Erdős-Rényi* (ER) random graphs. As the problem gets easier, instead, β increases and the marginals will progressively tend to one. In the extreme case of $\beta \to \infty$ (which is the easiest possible scenario), the magnetizations $\langle \sigma \rangle_\beta$ corresponds

A node partition of an ER random graph with $\mathscr{Q}^{\text{Mod}} \approx 0.25$



6 Recall that α is the parameter determining the hardness of CD in the DCSBM.

to the ground state configuration. The optimization approach hence boils down to the (overly optimistic) assumption that the ground-truth label vector corresponds to the minimum of the Hamiltonian, completely disregarding the contribution of the noise. This leads, in general, to over-fitting [ZK16], explaining why the modularity maximization leads to high scores on completely random graphs.

This concludes our introduction to the DCSBM and more generally to CD. We now proceed with a short review of some of the most influential methods for SC for CD, including both those inspired by the optimization and the Bayesian approaches. The literature on SC is incredibly vast and the following review considers a limited sample of all the existing contributions. The main criterion motivating the inclusion of a particular algorithm in the following section concerns whether or not it is equipped with a deep theoretical and algorithmic explanation.

4.3 SPECTRAL CLUSTERING: RELATED WORKS

Spectral clustering is a popular method to perform CD and has deep relations with both the optimization and the Bayesian approaches. Algorithm 4.1 describes the typical steps of SC for CD. Not all SC algorithms can be written in the form of Algorithm 4.1 that has the sole purpose of providing the reader a practical reference.

Taking a closer look at Algorithm 4.1, first note that k is required as an input. It has been extensively discussed that determining a good value for k is one of the big challenges of CD and SC is not an exception.

"Algorithm 0" of SC

Secondly, Algorithm 4.1 relies on the fact that some of the eigenvectors of an appropriate graph matrix representation M are *informative* with re-

Algorithm 4.1 : Typical SC algorithm on a graph with k classes

Input : $\mathcal{G}(\mathcal{V}, \mathcal{E}), k$
Output : Estimated label community vector $\hat{\boldsymbol{\ell}}$.

```
begin
    Choose the matrix M ∈ ℝ^{n×n}, a suited graph representation;
    Find the k largest (or smallest) eigenvalues of M and stack the
      corresponding eigenvectors in the columns of a matrix
      X ∈ ℝ^{n×k} ;
    (Optional) Normalize the rows of X ;
    Estimate community labels ℓ̂ as the output of k-means performed
      on the points in ℝ^k defined by the rows of X ;
    return ℓ̂.
end
```

Algorithm 4.2 : Lloyd algorithm for *k-means* [Llo82]

Input : Input data matrix $X \in \mathbb{R}^{n \times d_{\text{emb}}}$; k: number of clusters
Output : Label assignment $\hat{\boldsymbol{\ell}} \in \{1, \dots, k\}^n$

begin
  Initialize the vector $\boldsymbol{\ell}$;
  Set $t \leftarrow 0$ and $\boldsymbol{\ell}^{(t)} \leftarrow \boldsymbol{\ell}$;
  while *Convergence is not met* **do**
    $\boldsymbol{\mu}_a^{(t)}$ as in Equation (4.13) for $1 \leq a \leq k$;
    $\ell_i^{(t+1)} \leftarrow \arg\min_{a=1,\dots,k} \|X_{i,\bullet} - \mu_a^{(t)}\|^2$ for $1 \leq i \leq n$;
    $t \leftarrow t + 1$
  end
  return $\boldsymbol{\ell}^{(t-1)}$.
end

spect to the class structure of the graph. The arguments motivating and interpreting the existence of such eigenvectors are treated in the following sections and ultimately characterize each SC algorithm. Note that, from a practical viewpoint, the fact that these informative eigenvectors are in largest or smallest position is of fundamental importance, otherwise one would need to compute all eigenvectors and choose the most informative ones, with a very time consuming routine. However, as it will be made clear in the following, having k informative eigenvectors in dominant position cannot be given for granted, especially on sparse graphs.

The choice of the matrix M is perhaps the most fundamental task to design a SC algorithm. The small dimensional embedding X obtained from M must be a suited input to the final small dimensional clustering step. Coherently with the majority of the SC literature [CZC18], we systematically perform this step with the *k-means* algorithm, which we now summarize.

The k-means *algorithm*

Given the input matrix X, *k-means* assigns the labels $\hat{\boldsymbol{\ell}}$ as the solution to an optimization problem formulated as follows:

$$\hat{\boldsymbol{\ell}} = \arg\min_{\boldsymbol{\ell}} \sum_{a=1}^{k} \sum_{i \in \mathcal{V}_a^{\boldsymbol{\ell}}} \|X_{i,\bullet} - \boldsymbol{\mu}_a\|^2 \tag{4.12}$$

$$\boldsymbol{\mu}_a = \frac{1}{|\mathcal{V}_a^{\boldsymbol{\ell}}|} \sum_{i \in \mathcal{V}_a^{\boldsymbol{\ell}}} X_{i,\bullet}, \tag{4.13}$$

where $\mathcal{V}_a^{\boldsymbol{\ell}} = \{i \ : \ \ell_i = a\}$ is the set of all points with attributed label equal to a and $X_{i,\bullet}$ is the i-th row of X. The term $\boldsymbol{\mu}_a$ is the average of all the rows with label a and is the centre of the cluster. Equation (4.12) attributes label a to node i if $\boldsymbol{\mu}_a$ is the closest centre to $X_{i,\bullet}$. Note that, although not denoted explicitly, the term $\boldsymbol{\mu}_a$ depends on $\boldsymbol{\ell}$. This in is an NP-hard optimization problem for which an approximate solution is typically searched using Lloyd algorithm [Llo82], detailed in Algorithm 4.2. The labels are initialized (for example with a random assignment or with some more sophisticated

technique [AV06]) then the centres and the labels are iteratively updated, assigning each node to the closest centre. The complexity of Lloyd algorithm scales as $O(nkd_{\text{emb}})$ [SMR08].

With Lloyd's algorithm, there is no guarantee that the optimum of Equation (4.12) is reached and the output will depend, in general, on the initialization $\boldsymbol{\ell}$ (step 2 of Algorithm 4.2). To cure this flaw, *k-means* is commonly run multiple times and the solution reaching the lowest cost in terms of Equation (4.12) is kept.

Furthermore, note that the *k-means* objective function only involves the distances from the centres of each clusters. As a consequence, in the context of SC for CD, a good input for *k-means* is so that the distance between the points in the embedded space ideally depends only on the node's label. As we will comment in the following, however, the presence of a heterogeneous degree distribution must be accounted for, because it typically takes a (deleterious) role in determining the embedding.

With the basics on *k-means* being established, we now turn to a detailed description and justification of some state-of-the-art SC algorithms for CD.

4.3.1 CLASSICAL METHODS

The first contribution to SC for CD dates back to [Fie73] in which, for $k = 2$ communities, it was proposed to reconstruct communities using the eigenvector corresponding to the second smallest eigenvalue of the *combinatorial graph Laplacian* matrix $L = D - A$. It was then shown [VL07, for instance] that this eigenvector provides a relaxed solution of the RCut problem. In particular, let $\boldsymbol{f}$ be defined as follows

$$(\boldsymbol{f}_{\boldsymbol{\ell}})_i = \begin{cases} \sqrt{\frac{|\mathcal{V}_2|}{|\mathcal{V}_1|}} & \text{if } \ell_i = 1 \\ \sqrt{\frac{|\mathcal{V}_1|}{|\mathcal{V}_2|}} & \text{if } \ell_i = 2. \end{cases} \tag{4.14}$$

SC provides a relaxation of the graph-cut optimization problems

Then, it can be shown [VL07] that

$$\mathscr{Q}_A^{\text{RCut}}(\boldsymbol{\ell}) \propto \boldsymbol{f}_{\boldsymbol{\ell}}^T (D - A) \boldsymbol{f}_{\boldsymbol{\ell}}. \tag{4.15}$$

The relaxation consists in optimizing Equation (4.15) considering $\boldsymbol{f}_{\boldsymbol{\ell}} \in \mathbb{R}^n$, instead of having binary entries as in Equation (4.14). Then the labels can be estimated taking the sign of $\boldsymbol{u}$, the eigenvector associated to the smallest eigenvalue of $L = D - A$, subject to the constraint $\boldsymbol{u} \perp \mathbf{1}_n$. This condition is imposed to prevent all nodes from being set in the same class. Luckily, $\mathbf{1}_n$ is an eigenvector of L, precisely the one associated to its smallest eigenvalue, hence $\boldsymbol{u}$ is the eigenvector associated to $\lambda_2^{\uparrow}(L)$, *i.e.* to the second smallest eigenvalue. The result can be generalized to $k > 2$. Referring to Algorithm 4.1, one has $M = L$, the informative eigenvectors correspond to the smallest eigenvalues and step 4 is not performed.

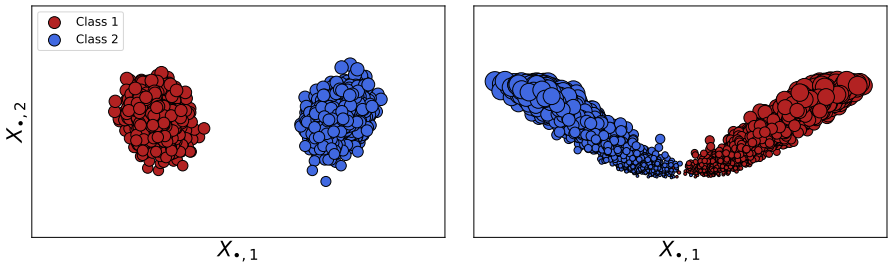


Figure 4.2: Scatter plot of the first vs second column of the matrix X containing the leading eigenvectors of $L^{\text{sym}} = D^{-1/2}AD^{-1/2}$. The colour code indicates the ground-truth labels, while the sizes of the points are proportional to the node degree. **Left**: homogeneous degree distribution. **Right**: heterogeneous degree distribution with $\theta_i \sim [\mathcal{U}(3,10)]^3$. For both plots $n = 5\,000$, $k = 2$.

Remark 4.3 (Multiplicity of the zero eigenvalue)**.** *It can be shown [VM10] that on a connected graph, $\lambda_2^{\uparrow}(L)$ (also known as* Fiedler eigenvalue*) is strictly positive. On a disconnected graph, instead, the multiplicity of the* 0 *eigenvalue equals the number of connected components of $\mathcal{G}(\mathcal{V},\mathcal{E})$, n_{cc} and the corresponding eigenvectors can be written as $v_i^{(1\leq a\leq n_{\text{cc}})} = \mathbb{I}_{i\in\mathcal{V}_a}$, where $\mathcal{V}_a$ are the nodes belonging to the a-th connected component. Running Algorithm 4.1 with $M = L$ and $k = n_{\text{cc}}$ returns a label assignment in which communities are identified by the connected components of the graph.*

A similar result justifies the exploitation of the *normalized graph Laplacian* matrices ($L^{\text{rw}} = D^{-1}A$ and $L^{\text{sym}} = D^{-1/2}AD^{-1/2}$) that provide a relaxed solution of the *NCut* problem [VL07]. In [SM00], $M = L^{\text{rw}}$, the informative eigenvectors correspond to the k largest eigenvalues and step 4 is again not performed. Similarly to Remark 4.3, the multiplicity of the eigenvalue 1 is equal to the number of connected components in $\mathcal{G}(\mathcal{V},\mathcal{E})$, in fact

$$(D - A)\boldsymbol{v} = 0 \implies D^{-1}A\boldsymbol{v} = \boldsymbol{v}.$$

The matrix L^{rw} is also called *random walk Laplacian* because its entry $L^{\text{rw}}_{ij} = \frac{1}{d_i}A_{ij}$ is the probability that a random walker transitions from node i to node j. The choice $M = L^{\text{sym}}$ is instead proposed in [NJW01a] in which the informative eigenvectors correspond to the k largest eigenvalues and step 4 is performed. Note that the matrices L^{rw} and L^{sym} have the same set of eigenvalues and, the eigenvectors are also trivially related, in fact

$$\begin{aligned} D^{-1}A\boldsymbol{x} &= \lambda\boldsymbol{x} \\ D^{-1}A\underbrace{D^{-1/2}\boldsymbol{y}}_{\boldsymbol{x}} &= \lambda\underbrace{D^{-1/2}\boldsymbol{y}}_{\boldsymbol{x}} \\ D^{-1/2}AD^{-1/2}\boldsymbol{y} &= \lambda\boldsymbol{y}. \end{aligned}$$

Strictly speaking, only the eigenvectors of L^{rw} and not those of L^{sym} provide a relaxed solution to the NCut problem. In several contexts, however, the use of L^{sym} is preferred because it is Hermitian.

From a graph-cut point of view, all three Laplacian matrices $L, L^{\text{sym}}, L^{\text{rw}}$ are able to create a node embedding which depends on the community labels and that thus justifies Algorithm 4.1 for the aforementioned choices of M. We recall, however, that a good embedding for *k-means* should also be independent from the degree distribution. Figure 4.2 compares the embedding obtained with the matrix L^{sym} in the case of a homogeneous degree distribution (on the left) and of a heterogeneous degree distribution (on the right). The size of each dot is proportional to the degree of the corresponding node, clearly evidencing that nodes with a small degree tend to be mapped to nearby points, potentially hampering the performance of *k-means*. It is for this reason that step 4 of Algorithm 4.1 is performed in [NJW01a], prior to *k-means*, so to obtain an embedding unspoiled by the degrees. This is not needed instead for the embedding obtained from L^{rw} that is naturally independent from the degree distribution [MRD21].

The effect played by the degrees in determining the small dimensional node embedding

Concluding, both the algorithms of [NJW01a] (based on L^{sym}) and of [SM00] (based on L^{rw}) consider and solve the problem induced by the presence of a heterogeneous degree distribution in the graph. On the opposite, the algorithm of [Fie73] (based on L) does not and its performance is consequently hampered on graphs with an arbitrary degree distribution.

Another classical choice of matrix M is in [New06], which is based on a relaxation of the modularity maximization problem. In this case M is the modularity matrix $M_{\text{mod}} = A - \frac{\boldsymbol{d}\boldsymbol{d}^T}{2|\mathcal{E}|}$. For $k = 2$ communities, the algorithm of [New06] finds the labels according to the sign of the leading eigenvector of M_{mod}, while for $k > 2$ the procedure is slightly more involved and it deviates from the general structure of Algorithm 4.1. The matrix M_{mod} is a rank one perturbation to the adjacency matrix, whose leading eigenvector is strongly correlated (in the sufficiently dense regime) to the degree vector:

Relaxation of the modularity maximization

$$(A\boldsymbol{d})_i = \sum_{j \in \partial i} d_j = d_i c + O(\sqrt{d_i}). \tag{4.16}$$

In the spectrum of M_{mod}, the leading eigenvector of A (corresponding to a *ferromagnetic* configuration in which all entries have the same sign) corresponds to an eigenvalue close to zero. All other eigenvectors, being approximately orthogonal to $\boldsymbol{d}$, are almost unperturbed. Consequently, the leading eigenvector of M_{mod} is strongly related to the eigenvector associated to the second largest eigenvalue of A.

The choice $M = A$ is thoroughly studied in [LR+15] in which SC is performed on the k leading eigenvectors of the adjacency matrix. To keep into account for the degree heterogeneity, a similar approach is considered in [Jin+15] for DCSBM-generated graphs. In this case, a modified version of step 4 in Algorithm 4.1 is performed, normalizing each row of X by $X_{i,1}$.

Many results justifying the earlier or more sophisticated SC algorithms exist in the *dense* regime [LR+15; CLV04; RCY+11; MRD21; AC17; GLM17b], also in the case for graphs with broad degree distributions. These are motivated by good concentration properties of the spectra of these matrices that

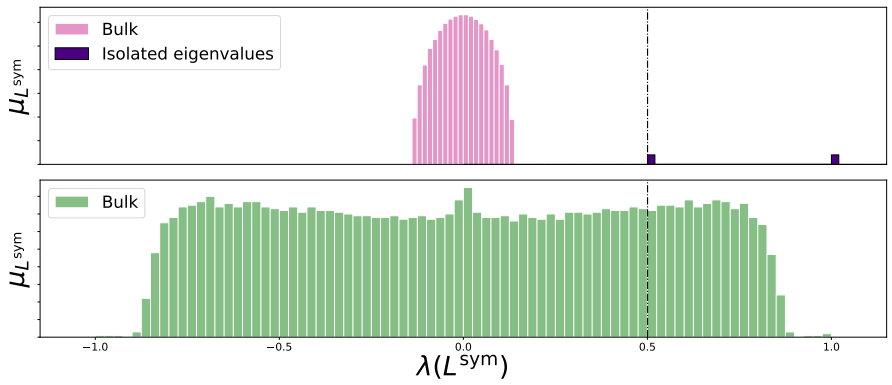
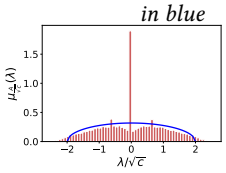


Figure 4.3: Empirical eigenvalue distribution of L^{sym} on SBM-generated graphs with $k = 2$. **Top**: dense regime, $c_{\text{in}} = 0.06n$, $c_{\text{out}} = 0.02n$. **Bottom**: sparse regime, $c_{\text{in}} = 6$, $c_{\text{out}} = 2$. For both simulations $n = 5\,000$.

can be predicted and studied with *random matrix theory* (RMT). However, as anticipated, real networks tend instead to be *sparse* and the aforementioned SC algorithms often perform poorly for practical applications. The origin of the problem if well understood comparing the two plots in Figure 4.3. In the top plot, the spectrum of L^{sym} (which we recall is the same as L^{rw}) is shown for a dense SBM-generated graph, clearly evidencing that the two largest eigenvalues are isolated. The bottom plot shows the spectrum of the same matrix for a sparse SBM with $c = O_n(1)$. In this case, the informative eigenvalue of L^{sym} is "swallowed" in the *bulk* of uninformative eigenvalues and running Algorithm 4.1 for $M = L^{\text{rw}}$ does not lead to a good community partition. The spreading of the bulk is not limited to the Laplacian matrices, but characterizes also the adjacency matrix. In [BGBK+19; BGBK+20] it was recently shown for the SBM and $c = o_n(\log(n))$ that the spectral measure of A does not tend to a limit distribution and that the largest eigenvalue correspond to the nodes with highest degree, and not to the community structure.

Spectrum of A for $c = O_n(1)$ compared to the limiting distribution in the dense regime in blue



It is important to underline that the spreading of the bulk is not only related to the asymptotic definition of sparsity. As a practical example, we consider the *political blogs* network [AG05] representing relations[7] between weblogs of USA politicians during 2004 elections. This network is particularly interesting as a benchmark for CD because a ground-truth label assignment can be defined according to whether the politician is liberal or democrat. Running the algorithms of [SM00] or [NJW01a] fails completely in recovering the ground truth assignment, since the vector $\boldsymbol{x}_2$ is completely uninformative, as shown in the left plot of Figure 4.4. It was however noticed in [JY+16] that the eigenvector $\boldsymbol{x}_3$ (associated to the third largest eigenvalue) is instead strongly correlated to the ground truth labels, as shown in the right plot of Figure 4.4. This is a practical example in which the informative eigenvector is swallowed in the uninformative bulk. In this particular case it is still easy to find but in more general settings, losing track of its

7 Note that this network is undirected but in Figure 4.4 we consider a symmetrized version of the adjacency matrix.

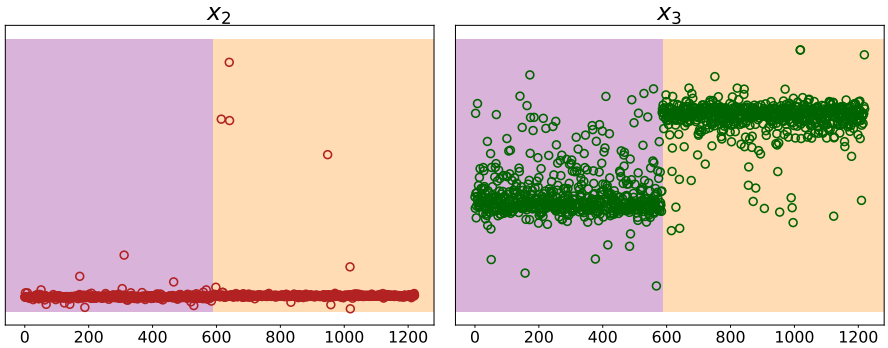


Figure 4.4: Eigenvectors of $L^{\text{rw}} = D^{-1}A$ on the (symmetrized) *political blogs* network [AG05]. **Left**: eigenvector associated to the second largest eigenvalue; **Right**: eigenvector associated to the third largest eigenvalue. The background color indicates the ground-truth labels.

precise position is a practical obstacle for SC. Besides, losing the isolation of informative eigenvalues, the associated eigenvectors tend to merge with the eigenvectors associated to close-by (non-informative) eigenvalues.

4.3.2 SPECTRAL CLUSTERING IN SPARSE GRAPHS

Finding efficient algorithms for SC in sparse graphs has been a challenge to which the scientific community has devoted a lot of attention in the latest years. One prominent line of research studied the positive effect that regularization has on SC. In particular, it was observed that using $D_\tau = D + \tau I_n$ and $A_\tau = A + \tau \mathbf{1}_n \mathbf{1}_n^T$ in place of D and A, much better concentration properties are achieved in sparse graphs and informative eigenvalues are restored to their dominant position [Ami+13; JY13; LR+15; LLV18; LLV15]. A particularly interesting method in this direction is proposed in [QR13] which uses the *regularized symmetric reduced Laplacian* matrix $L_\tau^{\text{sym}} = D_\tau^{-1/2} A D_\tau^{-1/2}$, for the *heuristic* choice $\tau = \bar{d}$, the average degree. In terms of Algorithm 4.1, the algorithm proposed in [QR13] sets $M = L_\tau^{\text{sym}}$, searches for the k eigenvectors corresponding to the k largest eigenvalues of M, and then performs the normalization step 4 on the rows of the resulting matrix X to cope with an arbitrary node degree distribution. In [ZR18] it was shown that also the algorithm of [QR13] can be understood as a relaxation of an optimization problem, called *CoreCut*.

Regularized SC

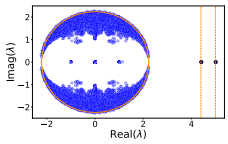
Generally speaking, the aforementioned works show that regularization improves the concentration properties of random matrices with low average degree. In all cases, however, the authors consider a *moderately sparse regime* in which the average degree goes (asymptotically) to infinity at a slow rate. The main theorems of these papers hence do not hold in the truly sparse regime in which $c = O_n(1)$. Besides, theory often suggests that large values of the regularizer τ should be preferred to small ones but small values of τ seem to perform better in practice [JY13; QR13]. In any case, regard-

less the lack of full theoretical justification, the algorithm of [QR13] can be considered by all means as state-of-the-art in terms of performance, dealing with both sparsity (meant in this case as low average degree) and the broad degree distribution that typically characterizes real graphs.

Statistical physics methods for CD

The earliest contributions to the theory of SC in *truly* sparse graphs came from statistical physics intuitions. In particular, the authors of [Krz+13] proposed an algorithm based on the non-backtracking matrix (Definition 1.8) for detection in the SBM. We recall from Chapter 2 that the matrix B is obtained from the linearization of BP equations that provide, in this case, the asymptotically exact solution to inference in the SBM. The algorithm of [Krz+13] cannot be expressed exactly in terms of Algorithm 4.1, since the size of B is $|\mathcal{E}_d|$ and not n. However, the p-th column of $X \in \mathbb{R}^{n\times k}$ is obtained from the vector $\boldsymbol{\psi}(\boldsymbol{g}_p)$, where $\boldsymbol{g}_p$ is the eigenvector of B associated to the p-th largest eigenvalue and $\big(\boldsymbol{\psi}(\boldsymbol{g}_p)\big)_i = \sum_{k\in\partial i} g_{p,ik}$.

The spectrum of B



From Claim 3.1, the authors of [Krz+13] conjectured that, if $\alpha > \alpha_c$ (Equation (4.10)), *i.e.* if there exists a polynomial-time algorithm capable of performing detection, then the vectors[8] $\{\boldsymbol{\psi}(\boldsymbol{g}_p)\}_{p=2,\dots,k}$ are correlated to the community structure and running *k-means* on the rows of X gives a label assignment with positive overlap. In order to show more clearly the origin of this claim, let us consider the case $k = 2$. According to Claim 3.1, the eigenvector of B associated to its second largest eigenvalue is correlated to the class structure, with $\mathbb{E}[g_{2,ij}] = \sigma_j$ and so is $\boldsymbol{\psi}(\boldsymbol{g}_2)$. The corresponding eigenvalue is $\lambda_2^{\downarrow|\cdot|}(B) = \frac{c_{\text{in}}-c_{\text{out}}}{2} + o_n(1)$, while the radius of the bulk is $\sqrt{c} + o_n(1)$. It is easy to check that the informative eigenvalue is isolated ($\lambda_2^{\downarrow|\cdot|}(B) > \sqrt{c}$) whenever $\alpha > \alpha_c$. This result is of fundamental importance because it defines a spectral algorithm in which the algorithmic and the detectability thresholds coincide. The spectral algorithm of [Krz+13] was the first (and still one of the few) obtaining this very powerful result. As we mentioned already, the conjectures of [Krz+13] have later been formally proved [Mas14] and extended also to the DCSBM setting [GLM16] in which the non-backtracking reaches as well the information-theoretic threshold.

A closely related algorithm is the one proposed in [SKZ14] which instead uses the eigenvectors attached to the smallest eigenvalues of the Bethe-Hessian matrix H_r. Recalling Proposition 3.1, the vector $\boldsymbol{\psi}(\boldsymbol{g}_p)$ used in [Krz+13] satisfies the following relation:

$$H_{\lambda_p^{\downarrow|\cdot|}(B)}\,\boldsymbol{\psi}(\boldsymbol{g}_p) = 0,$$

i.e. it is an eigenvector with zero eigenvalue of the Bethe-Hessian matrix for $r = \lambda_p^{\downarrow|\cdot|}(B)$. The matrix H_r has some important numerical advantage with respect to B: it is smaller in size and it is Hermitian. However, translating the algorithm of [Krz+13] in terms of the Bethe-Hessian requires to compute the k leading eigenvalues of B, annulling the advantage of using H_r. In [SKZ14]

8 Note that for $\boldsymbol{g}_{p=1}$ is the Perron-Frobenius eigenvector which is irrelevant for CD.

the authors determined a *unique* value of $r = \sqrt{\lambda_1^{\downarrow|\cdot|}(B)}$, guaranteeing that the k smallest eigenvalues of H_r are isolated for $\alpha > \alpha_c$. As a by product, the authors also showed that the number of negative eigenvalues of H_r for $r = \sqrt{\lambda_1^{\downarrow|\cdot|}(B)}$ is an estimator of the number of classes for the SBM. This is a very powerful result, because the algorithm of [SKZ14], unlike all others mentioned so far, does not need k as an input.

Remark 4.4. *Note that $\lambda_1^{\downarrow|\cdot|}(B)$ does not necessarily need to be calculated directly, but it can be estimated easily from the degree sequence $\boldsymbol{d}$ as follows, exploiting Theorem 3.2*

$$\lambda_1^{\downarrow|\cdot|}(B) = \frac{\sum_{i\in\mathcal{V}} d_i^2}{\sum_{i\in\mathcal{V}} d_i}.$$

Interestingly, the authors of [SKZ14] observed (with no further explanation) that SC based on their proposed Bethe-Hessian matrix performs slightly better than the non-backtracking matrix of [Krz+13], especially for $k > 2$.

All these results are powerful as they propose algorithms capable of reaching the information-theoretic threshold, but they also have inherent weaknesses as they only guarantee a positive correlation of their output classification with the underlying true structure. Specifically, these positive correlations do not imply that the classification performance is maximal. In particular, even in [GLM16] where spectral clustering on the DCSBM is studied, the problem of eigenvector pollution due to degree heterogeneity is not discussed and *a fortiori* not corrected.

4.4 CONTRIBUTIONS

In the state-of-the-art methods we presented, while sparsity is not properly accounted for in [Fie73; SM00; NJW01a; New06; LR+15; Jin+15], the line of research based on B (and consequently H_r) [Krz+13; SKZ14; BLM15; GLM16] does account for sparsity and provides methods to reach the detectability threshold; yet, all they guarantee is the possibility to obtain a positive, possibly suboptimal, correlation between the algorithm output and the underlying community structure. As for the works on regularization [Ami+13; JY13; LR+15; LLV18; LLV15; QR13], they only establish theoretical results of perfect community recovery far from the (most interesting) detection threshold.

In Chapter 5 we show that a proper parametrization of the Bethe-Hessian matrix allows one to solve both the issues of sparsity and heterogeneity at once on DCSBM-generated graphs. We propose a simple SC scheme which provably performs non-trivial clustering as soon as $\alpha > \alpha_c$ and is robust to degree heterogeneity, as it retrieves eigenvectors not infected by the node degrees. Our central result comes in the form of a claim which is supported by three parallel non-rigorous but convincing arguments. The proposed algorithm is extensively tested on synthetic DCSBM graphs and our simulations evidence that it outperforms standard competing spectral techniques.

In this chapter we further provide a natural explanation to the observed higher performances of the Bethe-Hessian of [SKZ14] with respect to the non-backtracking of [Krz+13].

Successively, Chapter 6 studies SC based on the matrix L_τ^{sym} and shows why a small value of τ is preferable, drawing a connection to our results of Chapter 5. We further show for which value of τ the leading eigenvectors of L_τ^{sym} allow for non-trivial community reconstruction as soon as $\alpha > \alpha_c$, addressing a further unanswered question of [QR13].

Chapter 7 then considers the problem of efficiently translating our theoretical results of Chapters 5, 6 into a practical algorithm which can be applied on arbitrary graphs. The proposed algorithm is extensively tested on real networks (for which the performance is measured in terms of modularity and DCSBM log-likelihood) and, in all cases, it outperforms or it is on par with the standard competing spectral techniques.

Finally, Chapter 8 closes Part II, providing a new vision to SC and in particular a compelling new connection between our proposed approach and all aforementioned standard spectral methods, so far treated independently. In particular, we show that our proposed algorithm for CD represents a "bridge" between the aforementioned competing methods and is capable to self-adapt to different levels of sparsity as well of hardness of the clustering problem.

5

OPTIMAL BETHE-HESSIAN FOR COMMUNITY DETECTION

Abstract

This chapter considers spectral community detection *(CD) on sparse graphs with heterogeneous degree distributions, generated from the* degree corrected stochastic block model *(DCSBM). We show that a conveniently parametrized sequence of Bethe-Hessian matrices* $\{H_{\zeta_p}\}_{p=1,\dots,k}$ *can be used to perform* spectral clustering *(SC) as soon as theoretically possible, without suffering from the graph degree heterogeneity. As opposed to competitive methods, our proposed parametrization inherently accounts for the hardness of the classification problem.*

Chapter 4 showed that in *spectral clustering* (SC) theory it is challenging to address simultaneously sparsity and heterogeneity in the degree distribution that typically characterize real-world graphs. Consequently, many commonly adopted spectral algorithms have poor performances on real datasets because either one of these two problems is not kept into account. This chapter shows that the eigenvectors corresponding to the smallest eigenvalues of the unweighted Bethe-Hessian matrix H_r, defined as

$$H_r = (r^2 - 1)I_n + D - rA, \tag{5.1}$$

The Bethe-Hessian matrix

can be exploited for a set of properly chosen values of r to efficiently perform SC in sparse graphs generated from the *degree corrected stochastic block model* (DCSBM) of Definition 1.11. This proposed parametrization naturally accounts for the hardness of the detection problem and we claim it to be optimal in the sense that no other parametrization (including the one of [SKZ14]) achieves higher performance, at least for DCSBM graphs. Moreover, for k classes of equal size and $C = c_{\text{out}}\mathbf{1}_k\mathbf{1}_k^T + (c_{\text{in}} - c_{\text{out}})I_k$, it obtains a positive overlap as soon as theoretically possible, achieving the DCSBM detectability threshold.

To the best our knowledge, there is no equivalent method which is explicitly adaptive to the hardness of the detection problem as the one described in this chapter. Moreover, the set of the optimal values of r can be directly and efficiently computed for *any* graph (hence not only for the DCSBM).

The central contribution of this chapter is a claim describing the shape of the eigenvectors attached to the smallest eigenvalues of the set of Bethe-Hessian matrices obtained for the optimal r values. This result comes in the form of a claim because, to the best of our knowledge, the mathematical tools to formally prove it are still lacking. The claim, however, is backed by several arguments as well as by numerous numerical simulations. For simplicity, we informally summarize here its content.

Informal statement of the main result

To each k-class graph, community detection is performed by first associating a set of k Bethe-Hessian matrices $\{H_{\zeta_p}\}_{p=1,\dots,k}$, with ζ_p defined so that the p-th smallest eigenvalue of H_{ζ_p} is equal to zero. We then extract from each matrix the eigenvector $\boldsymbol{x}_p$ attached to the zero eigenvalue ($H_{\zeta_p}\boldsymbol{x}_p = 0$). The eigenvectors $\{\boldsymbol{x}_p\}_{p=1,\dots,k}$ are stacked in the columns of the matrix X defined in Algorithm 4.1 and used to produce the small dimensional node embedding on which the *k-means* algorithm is applied. Claim 5.1 justifies the relevance of the above procedure as an efficient *community detection* (CD) method on DCSBM graphs, stating that:

- The largest value of ζ_p that can be defined is ζ_k. We show that, as a consequence, the $(k+1)$-th smallest eigenvalue of H_r is always positive for all r and the maximal number of negative eigenvalues of H_r as function of r is precisely equal to k. This allows one to build an estimator for the number of classes.

- The same hypotheses must be verified to guarantee that the k dominant eigenvectors of the non-backtracking matrix (Definition 1.8) and the set of vectors $\{\boldsymbol{x}_p\}_{p=1,\dots,k}$ bring information on the class structure. This, for k classes of equal size and $C = c_{\text{out}}\mathbf{1}_k\mathbf{1}_k^T + (c_{\text{in}} - c_{\text{out}})I_k$, implies that the vectors $\{\boldsymbol{x}_p\}_{p=1,\dots,k}$ can be used to reconstruct communities as soon as theoretically possible, i.e. when $\alpha > \alpha_c$.

- Unlike what was observed in Figure 4.2 where the eigenvectors of $L^{\text{sym}} = D^{-1/2}AD^{-1/2}$ (but similar comments may be made for A, $D - A$, H_r for a suboptimal choice of r) were tainted by the degree heterogeneity, the entries of $\{\boldsymbol{x}_p\}_{p=1,\dots,k}$ are not; consequently they provide a suited input for *k-means*.

The properties of $\{\boldsymbol{x}_p\}$ allow us to define an efficient algorithm for CD in sparse graphs with a heterogeneous degree distribution, solving at once the two main challenges discussed in Chapter 4. We now proceed with the formal statement of Claim 5.1, together with a clear definition of the hypotheses under which it is formulated.

5.1 MAIN RESULT

5.1.1 MODEL AND SETTING

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a k class DCSBM generated graph as in Definition 1.11. Recall that $C \in \mathbb{R}^{k\times k}$ is the class affinity matrix, while $\boldsymbol{\pi} \in \mathbb{R}^k$ is the vector determining the expected class sizes, with $0 < \pi_p < 1$ the probability of an arbitrary node to belong to class p. In order to consider an asymptotically non-trivial community detection setting, we work under Assumption 3.1, which we recall here for convenience.

Assumption 5.1. *Let $C \in \mathbb{R}^{k\times k}$ be the class affinity matrix and $\boldsymbol{\pi}$ be the expected class size vector as in Definition 1.11, with $\Pi = \mathrm{diag}(\boldsymbol{\pi})$. Further let $\boldsymbol{\theta}$ be the vector encoding the degree heterogeneity in the DCSBM and denote $\Phi = \mathbb{E}[\theta_i^2]$. The matrices C, Π satisfy the following hypothesis:*

- $C\Pi\mathbf{1}_k = c\mathbf{1}_k$, *for some positive* c,
- $\lambda_p(C\Pi)\Phi > \sqrt{c\Phi}$ *for all* $1 \leq p \leq k$.

In the particular case in which $\boldsymbol{\pi} = \mathbf{1}_k/k$ and $C = c_{\mathrm{out}}\mathbf{1}_k\mathbf{1}_k^T + (c_{\mathrm{in}} - c_{\mathrm{out}})I_k$, it is easily verified that $\lambda_p(C\Pi)\Phi > \sqrt{c\Phi}$ implies $\alpha > \alpha_c = 1$, *i.e.* we assume to be above the detectability threshold. In a more interesting setting in which communities have arbitrary sizes, Assumption 5.1 consists in assuming that the number of isolated eigenvalues of the non-backtracking matrix is equal to k. Note that, as we mentioned already in Chapter 3, Assumption 5.1 implies that, with high probability, the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ has a giant component, in agreement with Theorem 1.1.

Remark 5.1 (Negative eigenvalues of $C\Pi$). *It must be noted that Assumption 5.1 implies $\lambda_p(C\Pi) > 0$ for all $1 \leq p \leq k$. However, the results of Claim 5.1 remain valid if one replaces the second point of Assumption 5.1 by $|\lambda_p(C\Pi)\Phi| > \sqrt{c\Phi}$, that is, $\lambda_p^{\downarrow}(C\Pi)$ for $p \geq 2$ may be of arbitrary sign and only its modulus may be lower-bounded; $\lambda_1^{\downarrow}(C\Pi) = c$, being the Perron-Frobenius eigenvalue, is necessarily positive in any case.*

As the notations to describe this more general case are more cumbersome, we prefer here to focus on the simpler case where $C\Pi \succ 0$. Further comments to the more general setting in which $C\Pi$ may have negative eigenvalues will be made in the following. We now proceed providing the rigorous statement of our result.

5.1.2 CHARACTERIZATION OF THE INFORMATIVE EIGENVECTORS OF H_r

The central result of this chapter revolves around the characterization under Assumption 5.1 of a particular set of values $r = \zeta_p$ which must be carefully selected to best operate on H_r. These values are defined as follows.

Definition 5.1 (ζ_p). *Consider an arbitrary graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ composed of n_{cc} connected components. Let $\mathcal{G}^{(j)}(\mathcal{V}^{(j)}, \mathcal{E}^{(j)})$ be the subgraph reduced to the j-th connected component, and $H_r^{(j)}$ its associated Bethe-Hessian matrix. Let p be an integer between 1 and n. We define, if it exists, $\zeta_p^{(j)}$ as:*

The ζ_p's are well defined on an arbitrary graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$

$$\zeta_p^{(j)} = \min_{r \geq 1} \{ r : \lambda_p^{\uparrow}(H_r^{(j)}) = 0 \}. \tag{5.2}$$

In words, $\zeta_p^{(j)}$ is, if it exists, the smallest value of the parameter $r \geq 1$ such that the p-th smallest eigenvalue of $H_r^{(j)}$ is null.

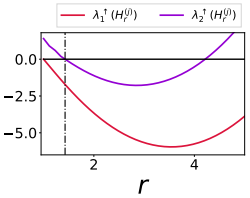
Note that, even though we focus in the remainder on DCSBM-generated graphs, the ζ_p's are well defined on arbitrary graphs. Here we list a few important properties of these values:

1. At $r = 1$, $H_1^{(j)} = D^{(j)} - A^{(j)}$ is the combinatorial Laplacian of the subgraph associated to the j-th connected component. As this subgraph is by definition connected, it is well known [CG97] that its smallest eigenvalue is null and that $\lambda_{p \geq 2}^{\uparrow}\left(H_1^{(j)}\right) > 0$. Consequently, $\zeta_1^{(j)} = 1$ always exists and if $\zeta_{p \geq 2}^{(j)}$ exists, it is strictly superior to 1.

2. If $\zeta_{p \geq 2}^{(j)}$ exists, then $\zeta_2^{(j)}, \ldots, \zeta_{p-1}^{(j)}$ necessarily exist and are smaller or equal to $\zeta_p^{(j)}$. In fact, if $\zeta_p^{(j)}$ exists, it means that at $r = \zeta_p^{(j)}$, 0 is the p-th smallest eigenvalue of $H_r^{(j)}$: the $p-1$ smallest are thus ≤ 0. Given that these $p-1$ smallest are ≥ 0 at $r = 1$ and by continuity of $\lambda_p^{\uparrow}(H_r^{(j)})$, they necessarily cross zero before $\zeta_p^{(j)}$. Similarly, if $\zeta_{p \geq 2}^{(j)}$ does not exist, then all $\zeta_q^{(j)}$ for $q > p$ do not exist.

3. Empirically, on as many connected graphs as we could think of, we have observed that the function $\lambda_p^{\uparrow}(H_r)$ for $r \geq 1$ either never crosses zero (in which case ζ_p does not exist), crosses zero exactly twice and is convex between these two crossings (in which case ζ_p is the lowest of the two values), or, in very symmetric cases, touches zero exactly once without crossing it (in which case ζ_p is that value).

Plot of $\lambda_{1,2}^{\uparrow}(H_r^{(j)})$. The vertical line is at $r = \zeta_2^{(j)}$.



With the definition of ζ_p being laid out, we are now in position to state our main result.

Claim 5.1. *Let $\mathcal{G}(\mathcal{V},\mathcal{E})$ be generated according to Definition 1.11, that is, a DCSBM with k classes. Let D and A be its degree and adjacency matrices, and $H_r = (r^2-1)I_n + D - rA$, for $r \in \mathbb{R}$, its Bethe-Hessian matrix. Provided that Assumption 5.1 is satisfied for $c = O_n(1)$, we have with high probability for all large n that:*

- *There is only one connected component j for which $\zeta_{p\geq 2}^{(j)}$ exists: it is the giant component. In the following, abusing notation, we simply write ζ_p instead of $\zeta_p^{(j)}$ to refer to the ζ's associated to this giant component. One has $\zeta_1 = 1$ and, if it exists, $\zeta_{p\geq 2}$ verifies*[1]

$$\zeta_p = \min_{r>1} \{r : \lambda_p^{\uparrow}(H_r) = 0\}. \tag{5.3}$$

- *The largest p for which ζ_p exists is equal to k, the number of underlying communities of the DCSBM.*[2] *One has $1 = \zeta_1 < \zeta_2 \leq \ldots \leq \zeta_k \leq \sqrt{c\Phi}$. More precisely:*

$$\forall\ p \text{ s.t. } 1 \leq p \leq k, \quad \zeta_p = \frac{c}{\lambda_p^{\downarrow}(C\Pi)} + o_n(1). \tag{5.4}$$

- *For $2 \leq p \leq k$ and $\zeta_p \leq r \leq \sqrt{c\Phi}$, the p smallest eigenvalues of H_r are isolated. In particular, zero is an isolated eigenvalue of H_{ζ_p}. Its corresponding eigenvector $\boldsymbol{x}_p$ is correlated to the community structure and the entries of $\boldsymbol{x}_p$ are in expectation (over realizations of A) independent of the degrees of the graph.*

In simple terms, Claim 5.1 predicts that, in a graph of k communities, SC can be successfully performed by successively retrieving the eigenvector associated to the null eigenvalue of each of the matrices $H_{\zeta_2}, \ldots, H_{\zeta_k}$. In the specific case of k classes of equal size with $C_{aa} = c_{\text{in}}$ and $C_{ab} = c_{\text{out}}$ if $a \neq b$, Claim 5.1 states that for $2 \leq p \leq k$ the ζ's are degenerate. As we will see next, this value of r differs from the choice $r = \sqrt{\lambda_1^{\downarrow|\cdot|}(B)} = \sqrt{c\Phi}$ advocated by [SKZ14], unless $\alpha = \alpha_c$, *i.e.* , exactly at the detectability threshold.

Figure 5.2 provides a visual representation of the typical spectrum of H_r for r not too far from a ζ_p: the eigenvalue of H_r closest to zero is clearly isolated and the eigenvector associated to this eigenvalue is strongly aligned to the community structure. In Figure 5.3, for a typical realization of a DCSBM, $\lambda_p^{\uparrow}(H_r)$ is represented as a function of r for $p = 1, 2, 3$ and 4 in solid lines

1 This statement allows to define ζ_p with respect to the Bethe-Hessian matrix of the whole graph, instead of the Bethe-Hessian matrix of its giant component. The fact that Equation (5.3) is verified with high probability is not evident. In fact, considering all the disconnected components at once might change the ordering of the eigenvalues. More details are to be found in Appendix B.1

2 Precisely, we will see that $\lambda_{k+1}^{\uparrow}(H_r) > 0$ for all $r > 1$ with high probability, so that ζ_{k+1} is not defined; in fact it was interestingly shown in [SKZ14] and we will verify that $\lambda_{k+1}^{\uparrow}(H_{\sqrt{c\Phi}}) \downarrow 0^+$ as $n \to \infty$, but the limit is never reached.

and $p = 5$ in dotted line. In this instance, ζ_5 does not exist (and all subsequent ζ_p's do not exist either), $\zeta_1 = 1$, and $\zeta_{2,3,4}$ exist and lie in $(1, 2)$.

The non-obvious parts of the claim are (i) of course that such ζ_p's do exist up to $p = k$, (ii) that they are concentrated around a deterministic value depending on the statistics of the model, (iii) importantly, that zero is indeed an isolated eigenvalue in the spectrum of H_{ζ_p}, (iv) that the associated eigenvector is informative for the underlying community structure and that is not infected by the degrees of the graph.

As we already mentioned, the statement of Claim 5.1 is formulated as a claim in the sense that, while efforts have been made to rigorously prove parts of the result [CZ20], the mathematical tools required to fully prove Claim 5.1, to the best of our knowledge, do not exist yet. Instead, the remainder of this section will propose three complementary supporting elements, arising from non-rigorous but convincing approximations, in particular borrowing arguments from the field of statistical physics. Specifically, we will successively show

- under Section 5.2.1 that the vectors $\boldsymbol{x}_p$, solution of $H_{\zeta_p}\boldsymbol{x}_p = 0$, are correlated with the community labels.
- under Section 5.2.2, that the informative null eigenvalue of H_{ζ_p} (associated to $\boldsymbol{x}_p$) is located in p-th smallest position and is isolated. This result is algorithmically crucial to determine ζ_p itself.
- under Section 5.2.3, that the entry $x_{p,i}$ is essentially independent of d_i, the degree of node i; precisely, it is strictly independent of d_i on average (over random allocation of the labels) and only depends on d_i through a noise term of order $1/\sqrt{d_i}$ otherwise.
- under Section 5.2.4 we briefly recap the proof of [CZ20] that considers the moderately sparse regime. We will in particular focus on both the main intuitions and weaknesses of the proof.

5.2 SUPPORTING ARGUMENTS

5.2.1 LINEARIZATION OF THE BP EQUATIONS

We first proceed to our agenda by arguing that the eigenvectors $\boldsymbol{x}_p$ of H_{ζ_p} are correlated with the community labels. Consider the *belief propagation* (BP) equations (4.7, 4.8, 4.9). These are the asymptotically Bayes optimal solution for inference in the sparse DCSBM. As we saw in Chapter 2 for the Ising model, these equations admit a trivial fixed point, for $\eta_{ij}(\ell_i) = \pi_{\ell_i}, \forall\ (ij) \in \mathcal{E}_{\mathrm{d}}$. It is a trivial solution because it leads to a random node partition. However, the non-backtracking matrix naturally comes into play linearising Equations (4.7, 4.8, 4.9) around this trivial fixed point.

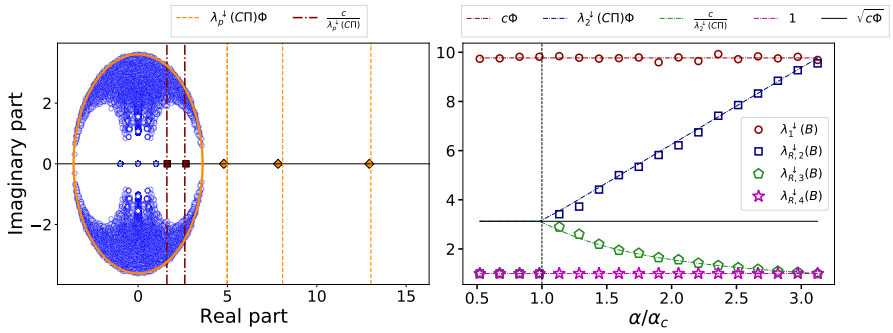


Figure 5.1: **Left**: spectrum of the matrix B on the complex plane. The blue dots represent the uninformative eigenvalues, the orange diamonds the real eigenvalues outside the bulk and the maroon squares the real eigenvalues inside the bulk. The lines indicate the theoretical values. For this network $n = 5\,000, k = 3, \theta_i \sim [\mathcal{U}(3,10)]^3, c = 8$. The off-diagonal elements of C are distributed according to $C_{p>q} \sim \mathcal{N}(c_{\text{out}}, c_{\text{out}}/k)$ with $c_{\text{out}} = 4$ and the diagonal elements are then fixed to have $C\Pi\mathbf{1}_k = c\mathbf{1}_k$. The element of $\boldsymbol{\pi}$ are distributed as $\pi_i \sim \mathcal{N}(1/k, 1/2k)$. The entries are then rescaled sot that $\text{Tr}(\Pi) = 1$. **Right**: position of the first four real eigenvalues of B as a function of α. For this simulation $n = 5\,000$, $k = 2$, $\theta_i \sim [\mathcal{U}(3,10)]^3$, $c = 6$.

In particular, let $\eta_{ji}(\ell_i) = \pi_{\ell_i} + b_{ji}(\ell_i)$ for small $b_\bullet(\cdot)$. Letting $\boldsymbol{b}^T = [\boldsymbol{b}_1^T, \dots, \boldsymbol{b}_k^T] \in \mathbb{R}^{|\mathcal{E}_d|k}$, with $\boldsymbol{b}_p \in \mathbb{R}^{|\mathcal{E}_d|}$ containing the entries of $b_{ij}(p)$, we obtain as in [Krz+13]:

Linearization of the asymptotically Bayes optimal BP equations

$$(T \otimes B)\boldsymbol{b} = \boldsymbol{b} + o_n(1), \tag{5.5}$$

where $\otimes$ is the Kronecker product and $T = \frac{\Pi C}{c} - \Pi\mathbf{1}_k\mathbf{1}_k^T$. From Equation (5.5), our interest is in the eigenvalues of $(T \otimes B)$ equal (or close) to one. From the properties of the Kronecker product, the set of eigenvalues of $(T \otimes B)$ is $\Lambda(T \otimes B) = \{\lambda_i(B)\lambda_j(T) : 1 \le i \le |\mathcal{E}_d|, 1 \le j \le k\}$. This induces a one-to-one relation between the $k-1$ non-zero eigenvalues[3] of T and the $k-1$ eigenvalues of B, that need to satisfy the relation $\lambda_q(T)\lambda_t(B) = 1$ for some t, q. From the expression of T, it comes that the p-th largest eigenvalue of T satisfies $\lambda_p^{\downarrow}(T) = \frac{1}{c}\lambda_{p+1}^{\downarrow}(C\Pi)$ for $1 \le p \le k-1$ (see Appendix B.2 for details) so that, for $2 \le p \le k$, there must exist $\zeta_p \in \Lambda(B)$ such that:

$$\zeta_p = \frac{c}{\lambda_p^{\downarrow}(C\Pi)} + o_n(1). \tag{5.6}$$

To these eigenvalues correspond $k-1$ *exact* eigenvectors $\boldsymbol{g}_p \in \mathbb{R}^{|\mathcal{E}_d|}$ of B satisfying $B\boldsymbol{g}_p = \zeta_p\boldsymbol{g}_p$ (with $\boldsymbol{g}_p = \boldsymbol{b}_p + o_n(1)$) that are naturally "informative" as they are associated to structural eigenvalues of T. Also, from a BP standpoint, these eigenvectors are small deviations from the uninformative fixed point, so must point towards informative directions.

3 Note that $\lambda_1^{\uparrow}(T) = 0$.

Note importantly here that Equation (5.6) *defines* ζ_p as a real eigenvalue of the matrix B. So far, this definition needs not correspond to the claimed definition of ζ_p as introduced in Claim 5.1: Section 5.2.2 will show that these two definitions are indeed equivalent. In addition, by Assumption 5.1, $\lambda_1^{\downarrow}(C\Pi) = c$ has unit multiplicity, so that $\zeta_p > 1 + o_n(1)$ for all $p > 1$. Along with Equation (5.6), we thus have $1 + o_n(1) < \zeta_p \leq \sqrt{c\Phi} + o_n(1)$, and thus ζ_p is asymptotically confined inside the bulk of radius $\sqrt{c\Phi} + o_n(1)$ of B. Figure 5.1 confirms that inside the disk of radius $\sqrt{c\Phi}$, these are the only informative eigenvalues of B. Specifically, the real eigenvalues of B inside the bulk are divided between (i) the (non-informative) eigenvalues $-1, 0, 1$, and (ii) the k eigenvalues $\lambda = \zeta_p \approx \frac{c}{\lambda_p^{\downarrow}(C\Pi)} \leq \sqrt{c\Phi}$ just described.

The ζ_p's are eigenvalues of B

Having identified the $k-1$ informative eigenvectors $\boldsymbol{g}_{2\leq p\leq k} \in \mathbb{R}^{|\mathcal{E}_d|}$ of B, we now recall that the embedding vectors the form the column of $X \in \mathbb{R}^{n\times k}$ appearing in Algorithm 4.1 are obtained from

$$\big(\boldsymbol{\psi}(\boldsymbol{g}_p)\big)_i = \sum_{k\in\partial i} g_{p,ki}.$$

The eigenvector attached to the null eigenvalue of H_{ζ_p} is informative

The vector $\boldsymbol{\psi}(\boldsymbol{g}_p)$ further satisfies

$$H_{\zeta_p}\boldsymbol{\psi}(\boldsymbol{g}_p) = 0,$$

hence $\boldsymbol{\psi}(\boldsymbol{g}_p) \equiv \boldsymbol{x}_p$. Note that, for $\zeta_1 = 1$ we have $H_1 = D - A$ and $\boldsymbol{x}_1$ is the vector $\mathbf{1}_n$, which is irrelevant to reconstruct communities.

Let us specifically focus on the particular case of k classes of equal size with the usual assumption $C_{ab} = c_{\text{in}}$ if $a = b$ and c_{out} otherwise. In this case efficient algorithms for CD exist only if $\alpha > \alpha_c$. Under this condition, B has two informative eigenvalues $\approx (c - c_{\text{out}})\Phi$ and $\approx \frac{c}{c-c_{\text{out}}}$ on either side of the disk (bulk) of radius $\sqrt{c\Phi}$. As the detection problem becomes harder, the two eigenvalues get closer together until they (asymptotically) meet right at the detectability transition $\alpha = \alpha_c$ where $(c - c_{\text{out}})\Phi = \sqrt{c\Phi}$. Further increasing the detection difficulty (that is, for $\alpha < \alpha_c$), the two eigenvalues become complex conjugate. This behavior is shown in Figure 5.1 (right panel) for $k = 2$. Summarizing, we have for $2 \leq p \leq k$

The ζ_p's are isolated eigenvalues of B if $\alpha > \alpha_c$

$$(c - c_{\text{out}})\Phi = \frac{c}{c - c_{\text{out}}} = \sqrt{c\Phi}, \quad \text{when } \alpha = \alpha_c$$
$$(c - c_{\text{out}})\Phi > \sqrt{c\Phi} > \frac{c}{c - c_{\text{out}}} > 1, \quad \text{when } \alpha > \alpha_c.$$

The fact that the leading eigenvalues of largest amplitude remain isolated down to the *detectability threshold* is the strongest argument in favor of the algorithm proposed by [Krz+13]. The authors in [Krz+13] however ignored the effect on the corresponding eigenvalue *inside* the bulk of B which, from our present discussion, is similar. This behavior can be extended to more than two classes with arbitrary sizes for which, however, the hypotheses of the conjecture of [Dec+11] do not hold.

Let us summarize these results in the form of the following argument.

Argument 5.1. *Under Assumption 5.1 for all large n with high probability, the non-backtracking matrix B of a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ generated from a k-class DCSBM has $k-1$ isolated real eigenvalues* inside *the disk of radius $\sqrt{c\Phi}$ (bulk). These eigenvalues are found at positions $1 < \zeta_p = \frac{c}{\lambda_p^{\downarrow}(C\Pi)} + o_n(1)$ for $2 \le p \le k$. Besides, the eigenvectors $\boldsymbol{g}_p$ of B associated with these eigenvalues ζ_p bring information about the community structure, which can be extracted through the vectors $\boldsymbol{x}_p \in \mathbb{R}^n$, defined as:*

$$x_{p,i} = \sum_{j\in\partial i} g_{p,ij}, \quad B\boldsymbol{g}_p = \zeta_p \boldsymbol{g}_p;$$

or equivalently satisfying

$$H_{\zeta_p}\boldsymbol{x}_p = 0.$$

Element 5.1 provides a first statement of Claim 5.1, according to which the class information should be retrieved from the vector $\boldsymbol{x}_p$ solution to $H_{\zeta_p}\boldsymbol{x}_p = 0$, and that $\zeta_p = c/\lambda_p^{\downarrow}(C\Pi) + o_n(1)$. This argument however does not specify the location (in the ordered list of the eigenvalues) of the null eigenvalue of H_{ζ_p} to which the eigenvector $\boldsymbol{x}_p$ corresponds nor the structure of the eigenvector $\boldsymbol{x}_p$, and in particular its dependence on the degrees of the graph. These aspects are covered in the subsequent sections.

5.2.2 THE ISING HAMILTONIAN ON $\mathcal{G}$

In this section, through a statistical physics mapping between Bayesian inference and an Ising model defined on the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, we aim to informally justify why the smallest eigenvalues of the matrix H_r correspond to "informative states" of the system and why the zero eigenvalue of H_{ζ_p} (associated by definition with the informative eigenvectors $\boldsymbol{x}_p$) should be isolated and in the p-th smallest position of the spectrum of H_{ζ_p}.

The smallest eigenvalues of H_r are informative.

In this section we provide a motivation on why the eigenvectors associated to the smallest eigenvalues of H_r should be the informative ones. For simplicity, let us first consider the case of $k = 2$ classes of equal size. We write the Ising Hamiltonian on the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$, with communities:

$$\mathcal{H}(\boldsymbol{s}) = -\sum_{(ij)\in\mathcal{E}} \operatorname{ath}\left(\frac{1}{r}\right) s_i s_j. \tag{5.7}$$

The term βJ has here been replaced by $\operatorname{ath}\left(\frac{1}{r}\right)$ for convenience. We now study the magnetization of this Hamiltonian with the (asymptotically exact) Bethe approximation detailed in Chapter 2. Let us recall the basics on the Bethe-Hessian matrix. The quantities $m_i = \langle s_i\rangle_\beta$ and $\chi_{ij} = \langle s_i s_j\rangle_\beta$ cannot be computed analytically on $\mathcal{G}(\mathcal{V},\mathcal{E})$, but are well approximated by

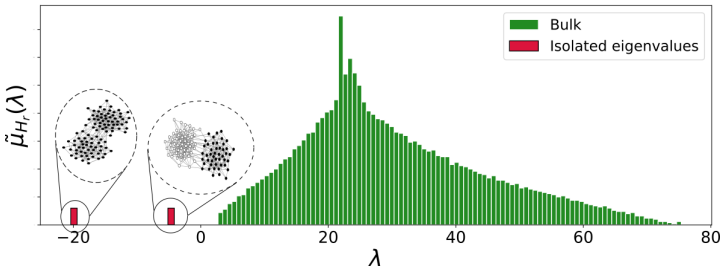


Figure 5.2: Spectrum of the matrix H_r for $r = 3/2\sqrt{c\Phi}$. On the x axis the eigenvalues, on the y axis the respective histogram. In green the bulk of uninformative eigenvalues and in red the two isolated eigenvalues – zoomed to simplify the readability –. The two drawings give a sketchy representation of the *ferromagnetic* ground state (all black spins) and of the informative state (half black and half white). For this network, $n = 5\,000$, $k = 2$, $\boldsymbol{\pi} = \mathbf{1}_k/2$, $c_{\text{in}} = 8$, $c_{\text{out}} = 3$, $r = 6.4$, $\theta_i \sim [\mathcal{U}(3,5)]^4$.

the *global* minimum of the Bethe free energy $\tilde{F}_\beta^{\text{Bethe}}(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\chi}})$, defined in Equation (2.17). For small values of β (or equivalently, for large r), the global minimum of $\tilde{F}_\beta^{\text{Bethe}}(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\chi}})$ is obtained for $\hat{\boldsymbol{m}} = \mathbf{0}_n$, while for smaller values of r, other local minima appear.

The appearance of new local minima can be obtained from the spectral properties of the Bethe-Hessian matrix H_r that corresponds to the Hessian of $\tilde{F}_\beta^{\text{Bethe}}(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\chi}})$ at $\hat{\boldsymbol{m}} = \mathbf{0}_n$.

$$\frac{\partial^2 \tilde{F}_\beta^{\text{Bethe}}(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\chi}})}{\partial \hat{m}_i \partial \hat{m}_i} = \frac{(r^2-1)I_n + D - rA}{r^2-1} \equiv \frac{H_r}{r^2-1}.$$

The Hessian in $\hat{\boldsymbol{m}} = \mathbf{0}_n$ is therefore trivially related to H_r. Negative eigenvalues in the spectrum of H_r correspond to the appearance of new local minima in $\tilde{F}_\beta^{\text{Bethe}}(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\chi}})$ that can be approximated by the corresponding eigenvectors. Recalling once again that for large r, all eigenvalues of H_r are positive, then the eigenvalues that may become negative (and are hence associated to a non-trivial magnetization) are the smallest ones. In Chapter 2 we already considered the *paramagnetic-ferromagnetic* phase transition in which there is one negative eigenvalue in the spectrum of H_r with the corresponding eigenvector having all positive entries. When $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a realization of the DCSBM with $k = 2$, however, also the second smallest eigenvalue of H_r may become negative: this is a consequence of Theorems 3.2 (describing the spectrum of B) and 2.1 (describing the relation between the eigenvalues of B and H_r). This eigenvalue corresponds to a further (local) minimum of the Bethe free energy which is induced by the community structure of $\mathcal{G}(\mathcal{V}, \mathcal{E})$. Let us summarize some important facts:

- A local minimum correlated to the class structure appears in $\tilde{F}_\beta^{\text{Bethe}}(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\chi}})$ because the configuration in which $s_i = 1$ if $\ell_i = 1$ and $s_i = -1$ if $\ell_i = 2$ is close to a local minimum of $\mathcal{H}(\boldsymbol{s})$: for two classes of equal size, minimizing $\mathcal{H}(\boldsymbol{s})$ with $\boldsymbol{s} \perp \mathbf{1}_n$ is equivalent to minimize the *RCut*.

- This informative minimum corresponds to the second smallest eigenvalue of H_r, because the ferromagnetic configuration has always a lower energy and it corresponds to the global minimum of $\tilde{F}_\beta^{\text{Bethe}}(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\chi}})$ at sufficiently low temperature.

The eigenvector associated to the second smallest eigenvalue of H_r is informative

- The interpretation of the eigenvectors of H_r is rigorous for large values of r (*i.e.* , at the paramagnetic point), but a similar behavior is expected to be observed also for other values of r. In particular, the informative eigenvector generally depends on r but it is still correlated to the class structure and found in second smallest position.

This relation between H_r and the Ising Hamiltonian allows us to make a further comment on the choice $r = \zeta_p$. Recall from Equation (4.11) that exact Bayesian inference can be performed for $k = 2$ studying the magnetization of the Hamiltonian

$$\mathcal{H}(\boldsymbol{s}) = -\sum_{(ij)\in\mathcal{E}} \frac{1}{2} \log\left(\frac{c_{\text{in}}}{c_{\text{out}}}\right) s_i s_j - \sum_{i\in\mathcal{V}} h_i(\boldsymbol{s}) s_i, \tag{5.8}$$

where we recall that $h_i(\boldsymbol{s})$ is a local field that prevents all node from being in the same class. From a straightforward calculation, we get

$$\frac{1}{2}\log\left(\frac{c_{\text{in}}}{c_{\text{out}}}\right) = \text{ath}\left(\frac{c_{\text{in}} - c_{\text{out}}}{c_{\text{in}} + c_{\text{out}}}\right) = \text{ath}\left(\frac{1}{\zeta_2 + o_n(1)}\right).$$

Relating Equation (5.7) to Equation (5.8), the choice $r = \zeta_2$ corresponds to setting at the Bayes optimal inference temperature. Yet, it should not be forgotten that in Equation (5.7) the local magnetic field does not appear. The consequence of neglecting this field is that the informative eigenvector of H_r does not correspond to the global (hence stable) minimum of $\tilde{F}_\beta^{\text{Bethe}}(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\chi}})$, but to a metastable local minimum. Discarding the field term is analytically convenient, but it creates an uninformative eigenvector in smallest position.

Let us now conclude this discussion, briefly commenting the case $k > 2$. It appears evident that the mapping between the Ising Hamiltonian and the Bayes optimal solution is, in this case, not straightforward. This is because labels, unlike spins, may take $k > 2$ different values. Nonetheless, a Ising Hamiltonian on $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with $k > 2$ is still useful for CD because in this case $k - 1$ local minima appear in the function $\tilde{F}_\beta^{\text{Bethe}}(\hat{\boldsymbol{m}}, \hat{\boldsymbol{\chi}})$. These minima are determined by the expectation of A and are, consequently, related to the community class and correspond to the k smallest eigenvalues of H_r.

We are thus left to showing why specifically the p-th smallest eigenvalue of H_r for the particular choice of $r = \zeta_p$ is of utmost importance, and why it corresponds to the null isolated eigenvalue of H_{ζ_p}.

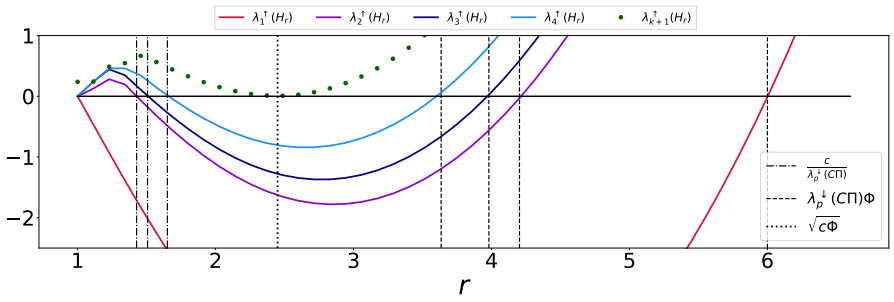


Figure 5.3: Behavior of $\lambda_p^\uparrow(H_r)$ as a function of r for $1 \leq p \leq k$ in solid lines, and for $p = k+1$ in dotted line. The vertical lines are the theoretical values of $c/\lambda_p^\downarrow(C\Pi)$(dashed dotted line), of $\sqrt{c\Phi}$ (dotted line) and of $\lambda_p^\downarrow(C\Pi)\Phi$ (dashed lines). For this simulation, $n = 50\,000$, $c = 6$, $c_{\text{out}} = 2$, $k = 4$, $C_{p>q} \sim \mathcal{N}(c_{\text{out}}, 2c_{\text{out}}/k)$ and $\pi \propto \mathbf{1}_k$.

Zero is an isolated eigenvalue of H_{ζ_p}.

This result follows from the observation, reported in Figure 5.3, that the k-th smallest eigenvalues $\lambda_1^\uparrow(H_r), \ldots, \lambda_k^\uparrow(H_r)$ of H_r successively equal zero in this order as r increases. Starting from $r = 1$, $H_r = H_1 = D - A$ for which we know that $\lambda_j^\uparrow(H_1) = 0$ for $1 \leq j \leq n_{\text{cc}}$ with n_{cc} the number of connected components and $\lambda_i^\uparrow(H_1) \geq 0$, $i > n_{\text{cc}}$. Increasing r beyond $r = 1$, the successive smallest eigenvalues of H_r first all increase and remain equal to the left edge of the bulk before escaping, each in turn (and in the order $\lambda_2^\uparrow(H_r), \ldots, \lambda_k^\uparrow(H_r)$), the bulk of non-informative eigenvalues. At their point of escape, they successively shift until they cross zero: this is where the successive values $\zeta_2 \leq \ldots \leq \zeta_k$ are defined. This in particular implies that the p-th smallest eigenvalue of H_{ζ_p} coincides with (i) the null eigenvalue of H_{ζ_p}, as well as with (ii) its largest isolated (so the last informative) eigenvalue. This allows us to redefine ζ_p as in Equation (5.3) of Claim 5.1. Note in passing that, letting r further increase beyond ζ_k, the left edge of the non-informative bulk of H_r progressively shifts back (from positive values) towards zero until it reaches asymptotically zero in the limit where $r = \sqrt{c\Phi}$ [SKZ14], before increasing again away for $r > \sqrt{c\Phi}$. These last findings may be summarized as follows.

ζ_p is an isolated eigenvalue of B and 0 is an isolated eigenvalue of H_{ζ_p}

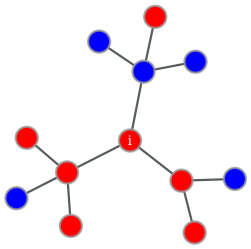
Argument 5.2. *Given a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ with k classes, generated from the DCSBM, the p smallest eigenvalues of the matrix H_r are isolated for $\zeta_p \leq r \leq \sqrt{c\Phi}$, for all $2 \leq p \leq k$. The entries of the p smallest eigenvectors are correlated with the class labels. Besides, in the specific case where $r = \zeta_p$, the p-th smallest eigenvalue of H_{ζ_p} is equal to zero.*

Argument 5.2 answers the problem of locating the informative eigenvalue-eigenvector pair $(0, \boldsymbol{x}_p)$ of H_{ζ_p} introduced in Argument 5.1, and of justifying why it is isolated. The last step, developed in the next section, consists in understanding the structural content of the informative eigenvector $\boldsymbol{x}_p$.

5.2.3 PARAMETRIZATION TO PROVIDE RESILIENCE TO DEGREE HETEROGENEITY

From a purely algebraic standpoint, the Bethe-Hessian matrix H_r may be seen as a regularized combinatorial graph Laplacian. In particular, we now show that $r = \zeta_p$ is the good parametrization for which the informative eigenvectors are resilient to degree heterogeneity. Our argument exploits the local convergence of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ to a *Galton-Watson* (GW) tree, that has been extensively commented in 1.3.2.

The labelled GW tree



Fixing A (and thus the degrees d_i), we consider the probability distribution of node label ℓ_i, conditionally to the label of one of its neighbours ℓ_j:

$$\begin{aligned}\mathbb{P}(\ell_i|\ell_j, A_{ij}=1) &= \frac{\mathbb{P}(\ell_i,\ell_j|A_{ij}=1)}{\mathbb{P}(\ell_j|A_{ij}=1)} = \frac{\iint d\theta_i d\theta_j \mathbb{P}(\ell_i,\ell_j,\theta_i\theta_j|A_{ij}=1)}{\mathbb{P}(\ell_j)} \\ &= \frac{\iint d\theta_i d\theta_j \mathbb{P}(A_{ij}=1|\theta_i,\theta_j,\ell_i,\ell_j)\mathbb{P}(\ell_i)\mathbb{P}(\ell_j)\mathbb{P}(\theta_i)\mathbb{P}(\theta_j)}{Z\pi_{\ell_j}} \\ &= \frac{\pi_{\ell_i} C_{\ell_i,\ell_j}}{c} = \frac{(\Pi C)_{\ell_i,\ell_j}}{c} = \frac{(C\Pi)_{\ell_j,\ell_i}}{c}. \end{aligned} \tag{5.9}$$

Let $\boldsymbol{v}_p$ be the eigenvector associated to the p-th largest eigenvalue of $C\Pi$ and let us define $\boldsymbol{u}_p \in \mathbb{R}^n$ with $u_{p,i} = v_{p,\ell_i}$ the n-dimensional class-wise expansion of $\boldsymbol{v}_p$: this vector $\boldsymbol{u}_p$ is inherently random as the class allocations $\ell_1, \dots, \ell_n$ are here considered random. Then, for n large, under the limiting tree approximation with conditionally independent offsprings, we take an expectation over the random allocation of the labels different from ℓ_i (denoted with $\boldsymbol{\ell}_{\setminus i}$), with A and ℓ_i known:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\ell}_{\setminus i}}[(A\boldsymbol{u}_p)_i|A,\ell_i] &= \sum_{j\in\partial i}\mathbb{E}_{\boldsymbol{\ell}_{\setminus i}}[u_{p,j}|A,\ell_i] = \sum_{j\in\partial i}\mathbb{E}_{\boldsymbol{\ell}_{\setminus i}}[v_{p,\ell_j}|A,\ell_i] \\ &\approx \sum_{j\in\partial i}\sum_{\ell_j}\mathbb{P}(\ell_j|\ell_i,A_{ij}=1)v_{p,\ell_j} = \frac{1}{c}\sum_{j\in\partial i}\sum_{\ell_j}(C\Pi)_{\ell_i,\ell_j}v_{p,\ell_j} \\ &= \frac{1}{c}\sum_{j\in\partial i}(C\Pi v_p)_{\ell_i} = \frac{d_i}{c}\lambda_p^{\downarrow}(C\Pi)v_{p,\ell_i} \\ &= d_i\frac{\lambda_p^{\downarrow}(C\Pi)}{c}u_{p,i} \end{aligned} \tag{5.10}$$

where the approximation follows from the fact that conditional independence of the neighbors of a given node only holds asymptotically. As a consequence of Equation (5.10), we find that

$$\mathbb{E}_{\boldsymbol{\ell}_{\setminus i}}[H_r\boldsymbol{u}_p|A,\ell_i] \approx \left[(r^2-1)I_n + D\left(1 - r\frac{\lambda_p^{\downarrow}(C\Pi)}{c}\right)\right]\boldsymbol{u}_p, \tag{5.11}$$

In order for this equation to be an approximate eigenvector equation for arbitrary degrees in D, the right hand-side term proportional to D must vanish. That is, one must select

$$r = \frac{c}{\lambda_p^{\downarrow}(C\Pi)} \approx \zeta_p$$

(this last approximation having been introduced and discussed in Section 5.2.1). This result implies that the eigenvectors $\boldsymbol{x}_p$ defined as $H_{\zeta_p}\boldsymbol{x}_p = 0$, for $1 \leq p \leq k$, correspond to a noisy version of $\boldsymbol{u}_p$ which is not affected *on average over the class allocation* by the degree distribution. The approximation holds beyond the average for every typical realization of the class allocation to first order in d_i. In particular, denoting $\boldsymbol{\xi}_p$ the "noise" vector satisfying $\boldsymbol{x}_p = \boldsymbol{u}_p + \boldsymbol{\xi}_p$, and thus $H_{\zeta_p}(\boldsymbol{u}_p + \boldsymbol{\xi}_p) = 0$, we get

The expectation of $\boldsymbol{x}_p$ is independent of the degree distribution

$$\begin{aligned}
\mathrm{Var}_{\boldsymbol{\ell}_{\setminus i}}[(H_{\zeta_p}\boldsymbol{u}_p)_i | A, \ell_i] &= \mathrm{Var}_{\boldsymbol{\ell}_{\setminus i}}[(H_{\zeta_p}\boldsymbol{\xi}_p)_i | A, \ell_i] \\
&= \mathrm{Var}_{\boldsymbol{\ell}_{\setminus i}}[(\zeta_p^2 - 1 + d_i)\xi_{p,i} - \zeta_p(A\boldsymbol{\xi}_{p,i}) | A, \ell_i] \\
&\approx O_{d_i}(d_i^2)\mathrm{Var}_{\boldsymbol{\ell}_{\setminus i}}[\xi_{p,i} | A, \ell_i] + O_{d_i}(d_i)\mathrm{Var}_{\boldsymbol{\ell}_{\setminus i}}[\xi_{p,i} | A, \ell_i] \\
&\approx O_{d_i}(d_i^2)\mathrm{Var}_{\boldsymbol{\ell}_{\setminus i}}[\xi_{p,i} | A, \ell_i]. \qquad (5.12)
\end{aligned}$$

In this derivation we used the fact that the random variables $\xi_{p,i}$ and $[A\boldsymbol{\xi}_p]_i$ are essentially independent and that the variance of the sum of the d_i asymptotically independent variables $\xi_{p,1}, \ldots, \xi_{p,n}$ grows linearly with d_i. Now, proceeding as in Equation (5.10), we can compute $\mathrm{Var}_{\boldsymbol{\ell}_{\setminus i}}[(H_{\zeta_p}\boldsymbol{u}_p)_i | A, \ell_i]$ from a direct calculation of $\mathbb{E}_{\boldsymbol{\ell}_{\setminus i}}[(H_{\zeta_p}\boldsymbol{u}_p)_i^2 | A, \ell_i]$ and $\mathbb{E}^2_{\boldsymbol{\ell}_{\setminus i}}[(H_{\zeta_p}\boldsymbol{u}_p)_i | A, \ell_i]$, and we obtain (to leading order in d_i) $\mathrm{Var}_{\boldsymbol{\ell}_{\setminus i}}[(H_{\zeta_p}\boldsymbol{u}_p)_i | A, \ell_i] = O_{d_i}(d_i)$.

Combining with Equation (5.12), we get that the variance scales approximately as d_i^{-1}. The vector $\boldsymbol{x}_p$ can therefore be written as the sum of the deterministic information and a noise with amplitude inversely proportional to the square root of the degree, consistently predicting that nodes with higher degrees are easier to classify.

These results are summarized under the form of our last argument.

Argument 5.3. *The eigenvector $\boldsymbol{x}_p$ ($1 \leq p \leq k$), solution to $H_{\zeta_p}\boldsymbol{x}_p = 0$, is a noisy version of the vector $\boldsymbol{u}_p$, defined as $u_{p,i} = v_{p,\ell_i}$, where $C\Pi\boldsymbol{v}_p = \lambda_p^{\downarrow}(C\Pi)\boldsymbol{v}_p$. The noise for entry i scales approximately as $1/\sqrt{d_i}$ for d_i sufficiently large and is zero on average. Consequently, the entries of $\boldsymbol{x}_p$ do not, to first order, depend on the degree distribution but only on the labels.*

Figure 5.4 confirms our argument: a $k = 2$ class DCSBM graph is considered and the informative eigenvector $\boldsymbol{x}_2$ is plot for $r = \zeta_2$ and $r = \sqrt{c\Phi}$ (as in [SKZ14]) with the marker size of the i-th entry being proportional to $1/\sqrt{d_i}$. We see that for a suboptimal choice of r, all nodes with a small degree are embedded to nearby points. For $r = \zeta_2$ instead, nodes with a small degree are more likely to be misclassified, but they are not embedded to nearby points. Moreover, the histograms of the eigenvector entries further confirm this observation. For the optimal regularization, two distinct clouds density peaks appear, while for $r = \sqrt{c\Phi}$ most eigenvector entries (in the example we chose a power law degree distribution) are close to zero.

The following remark summarizes the major consequence of this result.

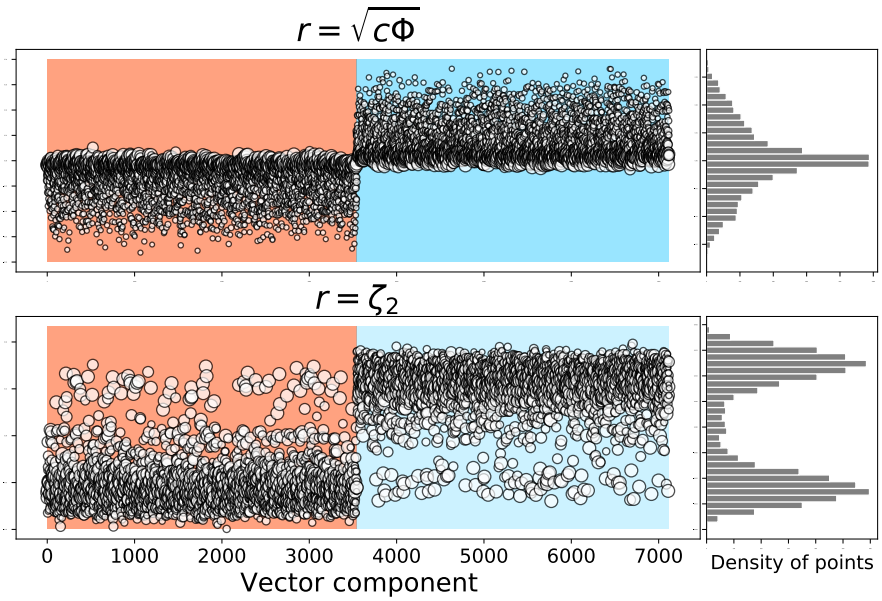
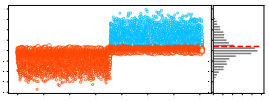


Figure 5.4: Informative eigenvector $\boldsymbol{x}_2$ of the Bethe-Hessian matrix for $r = \sqrt{c\Phi}$ (top) and $r = \zeta_2$ (bottom). On the right, the corresponding histogram. For this plot, $n = 10\,000$, $k = 2$, $\boldsymbol{\pi} = \mathbf{1}_2/2$, $c_{\text{in}} = 14$, $c_{\text{out}} = 2$, $\theta_i \sim [\mathcal{U}(3,10)]^6$. Only the giant component of $\mathcal{G}(\mathcal{V},\mathcal{E})$ is considered. The background colours refer to the ground-truth labels, while the marker-size is proportional to $1/\sqrt{d_i}$, where d_i is the degree of node i.

Remark 5.2 (Embedding quality and *k-means*)**.** *Spectral clustering has the role to produce a a suited input to some small dimensional clustering algorithm, such as* k-means*. Referring to Figure 5.4, the role of* k-means *is to draw a horizontal line on the y axis, that best separates the two clouds of points. This line is $y \approx 0$ (we considered two classes of equal size). Comparing the two histograms, one sees that, for $r = \sqrt{c\Phi}$, at 0 there is a density peak, while at $r = \zeta_2$ the density is minimal. For this reason, a slight deviation in the location of y leads to the misclassification of several nodes in the earlier case, but not in the latter. We can say that the optimal choice of r produces an embedding that is more robust to the* k-means *step. For pictorial purposes, Figure 5.4 limits to the case $k = 2$, but the same observation holds for $k > 2$ classes of arbitrary size in which the negative effect on* k-means *of a low quality embedding is even more exacerbated.*

The output of k-means *on $\boldsymbol{x}_2$ for $r = \sqrt{c\Phi}$*



Element 5.3 combined with Elements 5.1, 5.2 completes our justification of Claim 5.1. In the next section we give a further argument related to the existence and location of the ζ_p's that holds for moderately dense graphs generated from the *stochastic block model* (SBM).

5.2.4 ζ_p IN THE DENSE REGIME

This section briefly provides the main result of [CZ20] in which the authors attempted to prove parts of Claim 5.1. They focused in particular on the existence of isolated eigenvalues inside the bulk of B in the case of $k = 2$ communities of equal size. For simplicity, the authors of [CZ20] considered a sufficiently dense regime with $\log(n)/c = o_n(1)$ under the SBM assumption. As already discussed in Section 3.2.2 this setting is particularly convenient due to the degree concentration: $\|D - cI_n\| = o_n(c)$. This allows one to relate the eigenvalues of B and A. In particular, for $r \in \mathbb{C} \setminus \{\pm 1\}$ an arbitrary eigenvalue of B, there exists $\tilde{r} = r + o_n(r)$ satisfying

In the dense SBM setting, the eigenvalues of B are trivially related to those of A

$$\tilde{r} = \frac{\lambda_i(A) \pm \sqrt{\lambda_i^2(A) - 4(c-1)}}{2},$$

as it was already showed in Equation (3.7). For any $\lambda_i(A) > 4(c-1)$ one simply obtains that $\tilde{r}$ has two solutions (denoted with $\tilde{r}_\pm$) that satisfy

$$\begin{aligned} \tilde{r}_+ &= \lambda_i(A) + o_n(c) \\ \tilde{r}_- &= \frac{c}{\tilde{r}_+} + o_n(1). \end{aligned}$$

In particular, for the SBM and $1 \le i \le k$, one has that $\lambda_i^{\downarrow}(A) = \lambda_i^{\downarrow}(C\Pi) + o_n(c)$, confirming Claim 5.1 and naturally justifying the mapping between inner and outer isolated eigenvalues of B. The main technical contribution of [CZ20] is to prove that $r = \tilde{r} + o_n(r)$ and will not be further commented here. Albeit the proof [CZ20] provides a very natural explanation to the existence of isolated eigenvalues in the bulk of B, it strongly relies on the degree concentration assumption which does not hold in the sparse regime and, above all, for the DCSBM. As we commented already in Section 3.2.2, this is likely to be only a technical hypothesis since the empirical evidence suggests that the behavior of the isolated eigenvalues of B is essentially preserved in the presence of an arbitrary degree distribution.

5.3 PERFORMANCE COMPARISON

In this section, Claim 5.1 is translated into Algorithm 5.1 for SC and tested on synthetic graphs. In Algorithm 5.1, the values of $\zeta_{2 \le p \le k}$ are successively estimated and the eigenvector corresponding to the zero eigenvalue of H_{ζ_p} is stored in the columns of X. This matrix determines the small dimensional node embedding to on which *k-means* is applied.

Remark 5.3 (Value of k). *Note that Algorithm 5.1 requires k as an input. If not known in advance, k can be estimated counting the number of negative eigenvalues of $H_{\sqrt{c\Phi}}$, according to [SKZ14]. Further comments on how to efficiently implement this estimation will follow in Chapter 7.*

Algorithm 5.1 : Community Detection with the Bethe-Hessian

Input : adjacency matrix of undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, number of classes k

Output : Estimated label community vector $\hat{\boldsymbol{\ell}}$.

begin
  Estimate $c = \frac{1}{n}\sum_i d_i$ and $\Phi = \frac{1}{nc}\sum_i d_i^2$;
  for $p = 2 : k$ **do**
    $\zeta_p \leftarrow r$ such that $\lambda_p^{\uparrow}(H_r) = 0$ on $1 < r \leq \sqrt{c\Phi}$;
    $X_{\bullet,p} \leftarrow \boldsymbol{x}_p$ such that $H_{\zeta_p}\boldsymbol{x}_p = 0$;
  end
  Estimate community labels $\hat{\boldsymbol{\ell}}$ from the node embedding $X = [X_{\bullet,2}, \ldots, X_{\bullet,k}]$;
  return $\hat{\boldsymbol{\ell}}$.
end

Generally speaking, Algorithm 5.1 can be considered as a *meta-algorithm* that is written for user readability. Deeper considerations on how to efficiently implement it on arbitrary graphs (hence beyond the DCSBM) assumption will be dedicated in Chapter 7.

In Figure 5.5 we compare the overlap performance (defined in Equation 4.1), achieved by Algorithm 5.1 versus competing methods for synthetic DCSBM graphs as a function of the hardness of the problem α. On the left hand-side of Figure 5.5 is depicted the case $k = 2$, while on the right $k = 5$. In both cases detection is asymptotically feasible only if $\alpha > \alpha_c$.

In both cases, Algorithm 5.1 outperforms the competing spectral algorithms and has a very close performance to the Bayes optimal solution obtained with the BP algorithm[4] of [Dec+11]. As $\alpha \to \alpha_c$, Algorithm 5.1, the algorithm of [SKZ14] based on $H_{\sqrt{c\Phi}}$ and the one of [Krz+13] based on B give essentially the same result, which confirms that they are indeed equivalent at the phase transition. For easier detection problems though, except for the algorithm of [QR13] which is only slightly less accurate, the performance of all methods is largely improved by Algorithm 5.1. Interestingly, note that $|\sqrt{c\Phi} - \zeta_2| < |\lambda_2^{\downarrow}(C\Pi)\Phi - \zeta_2|$, giving an intuition on why the Bethe-Hessian method of [SKZ14] performs better than the non-backtracking approach of [Krz+13]. As for the standard spectral clustering algorithm which exploits the dominant eigenvectors of A [LR+15], it can only perform non trivial community reconstruction for α far beyond the threshold $\alpha = \alpha_c$. The algorithm of [SM00], based on the random walk Laplacian L^{rw}, is here incapable of making any non-trivial reconstruction for the considered set of parameters, suggesting that its dominant eigenvectors are not informative.

To further confirm our theoretical results, Figure 5.6 depicts the overlap of Algorithm 4.1 with $M = H_r$ for different values of r. For this simulation,

4 The codes for the SBM are taken from the first author of [Dec+11] personal webpage and are adapted to the DCSBM setting modifying the field term appearing in Equations (4.7, 4.8, 4.9).

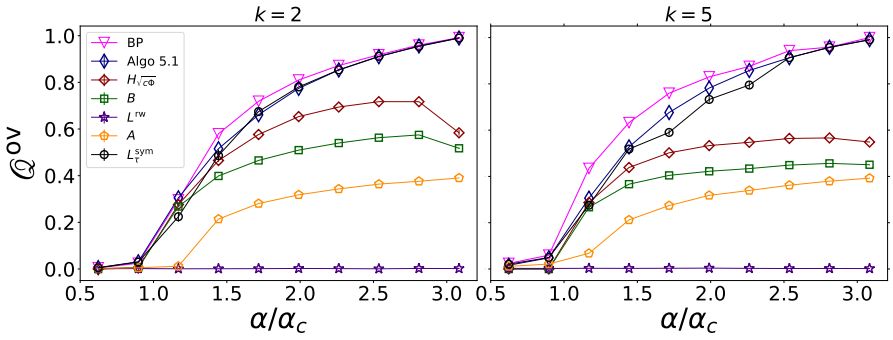


Figure 5.5: Overlap comparison as a function of the hardness α of the detection problem. Blue sharp diamonds are the result given by Algorithm 5.1, red diamonds the algorithm using the Bethe-Hessian of [SKZ14], green squares are for the non-backtracking of [Krz+13], yellow pentagons the adjacency matrix as in [LR+15], purple stars the random walk Laplacian, black dots are the algorithm of [QR13], the pink triangles are the Bayes optimal solutions obtained with BP [Dec+11]. For both graphs $n = 50\,000$, $\theta_i \sim [\mathcal{U}(3,10)]^4$, $\boldsymbol{\pi} \propto \mathbf{1}_k$, $c = 5$, $C = c_{\text{out}}\mathbf{1}_k\mathbf{1}_k^T + (c_{\text{in}} - c_{\text{out}})I_k$. Averages over 10 samples.

$k = 2$ classes of equal size are considered. When α is large enough, small values of r lead to better partitions than large values of r that are more affected by degree heterogeneity. However, for r small, the informative eigenvector is not necessarily corresponding to the second smallest eigenvalue, leading to a meaningless partition. On the contrary, larger values of r show isolated eigenvectors also in the "hard regime". We recall that $r = \zeta_2$ is an α-dependent parameter: for $\alpha \to \alpha_c$, ζ_2 is "large enough" so that the informative eigenvalue is isolated, while for $\alpha \gg \alpha_c$ it is "small enough" to give good partitions. Also the value of $r = (c_{\text{in}} - c_{\text{out}})\Phi/2$ is α-dependent and it corresponds to clustering with B as indicated in [Krz+13]. While it gives good partitions very close to the transition, this choice of r seems largely sub-optimal for easier tasks.

From a direct computation, Figure 5.6 hence suggests that the choice $r = \zeta_p$ is indeed optimal, in the sense that no other value of r achieves a better performance on DCSBM-generated graphs.

5.4 CONCLUSION

This chapter introduced the main theoretical result of this manuscript, formulated in the form of a claim under the DCSBM assumption. Several independent arguments as well as numerical tests foster the correctness of Claim 5.1. Summarizing, the key contributions of this chapter can be listed as follows.

- We show the existence of a parametrization of the Bethe-Hessian matrix that allows one to produce a node embedding resilient to the presence of a broad degree distribution. This embedding is obtained with-

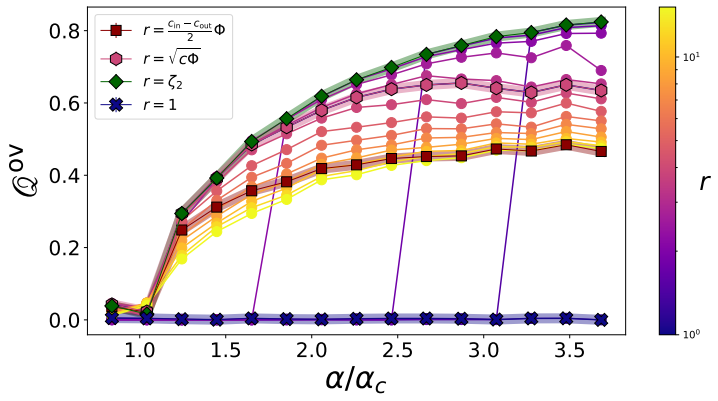


Figure 5.6: Overlap comparison as a function of α, using the second smallest eigenvector of H_r, for different values of r. In colour code the values of r ranging from $r = 1$ (blue) to $r = c\Phi$ (yellow). The red squares indicate $r = (c_{\text{in}} - c_{\text{out}})\Phi/2$, that is equivalent to clustering with the matrix B [Krz+13],the purple hexagons represent the Bethe-Hessian of [SKZ14], the green diamonds are the proposed Algorithm 5.1 and the blue crosses are the graph Laplacian ($r = 1$). For these simulations, $n = 20\,000$, $c = 8$, $\theta_i \sim [\mathcal{U}(3, 10)]^4$. Average over 10 realizations.

out any preprocessing on the matrix X appearing in Algorithm 4.1 and naturally discards the "noise" introduced by the degrees.

- The proposed parametrization is deeply related to the parameters of generative model and with the Bayes optimal inference solution. With Algorithm 5.1 it is possible to estimate the optimal parametrization in an efficient and completely unsupervised fashion.

- We claim, backed by several arguments and simulations, that the proposed parametrization is optimal, in the sense that no other parametrization of H_r leads asymptotically to better performances on graphs generated from the DCSBM.

All the results above have been thoroughly studied for an *assortative* community structure, in which each node has the largest probability to get connected to nodes in its own class. As suggested in Remark 4.3, however, all the results can be formulated in a disassortative setting. For instance, for $k = 2$ communities, let $c_{\text{in}} < c_{\text{out}}$. The corresponding graph has more edges connecting nodes in opposite communities that those connecting nodes in the same community and is, therefore, *disassortative*. The second eigenvalue of $C\Pi$ is $(c_{\text{in}} - c_{\text{out}})/2 < 0$ and consequently $\zeta_2 < 0$.

Disassortative graphs

Recalling the relation between CD and the Ising model, a negative value of ζ_2 consists in placing an *antiferromagnetic* interaction $J < 0$, favouring configurations in which neighbouring nodes have misaligned spins. In this case, the ferromagnetic configuration $\mathbf{1}_n$ would not correspond to the global minimum of the Bethe free energy and the informative eigenvector would be associated to the smallest (and not second smallest) eigenvalue of $H_{r<0}$.

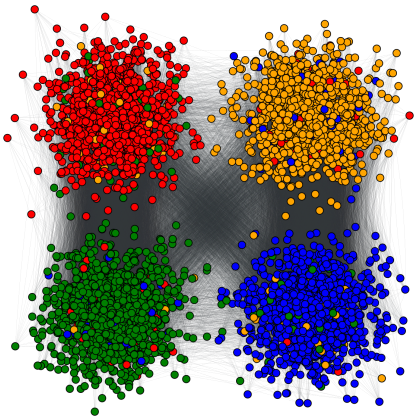


Figure 5.7: Plot of a SBM graph with $n = 5\,000$, $k = 4$ communities and C as in Equation (5.13). The colours are attributed according to the output of Algorithm 5.1, adapted to operate with negative values of r, while the nodes positions are attributed according to the ground truth labels.

More general cases can be designed in which both positive and negative eigenvalues appear in the spectrum of $C\Pi$. While the mathematical result is well defined, the interpretation of the negative eigenvalues of $C\Pi$ is not straightforward. As an example, the right frame of Figure 5.7 displays a network designed from the SBM with $k = 4$, $\Pi \propto I_k$ and C defined as:

$$C = \begin{pmatrix} c_{\text{in}} & c_{\text{out}} \\ c_{\text{out}} & c_{\text{in}} \end{pmatrix} \otimes \begin{pmatrix} c_{\text{out}} & c_{\text{in}} \\ c_{\text{in}} & c_{\text{out}} \end{pmatrix} \tag{5.13}$$

with '$\otimes$' the Kronecker product. This graph has a *hierarchical* structure: it can be divided into two assortative communities, each of them composed of two disassortative communities. In Figure 5.7 the position of each node is assigned according to its true label, while the color is assigned according to the output of k-means on the vectors $\boldsymbol{x}_p$ solution to $H_{\zeta_p}\boldsymbol{x}_p = 0$. We can observe that our algorithm performs well also in this particular case.

The results of this chapter introduce some relevant questions that will be answered in the following. In particular, Figure 5.5 shows that the algorithm of [QR13] performs very competitively with Algorithm 5.1. The theory justifying the algorithm of [QR13], however, does not cover neither the case of sparse graphs with $c = O_n(1)$, nor it provides any comment on the possibility of this algorithm to perform non trivial clustering down to the theoretical *detectability threshold* α_c. The high performance achieved by [QR13], however, requires a deeper understanding of the role of regularization in SC.

Chapter 6 brings new conclusions in this direction working on an explicit relation between the Bethe-Hessian and regularized Laplacian matrices. We will show, in particular, that the regularized Laplacian matrix of [QR13] can indeed be exploited to achieve non-trivial clustering as soon as $\alpha > \alpha_c$ in the sparse regime $c = O_n(1)$, yet for a slightly different value of the regularizer τ with respect to the one proposed in [QR13].

6

OPTIMAL LAPLACIAN REGULARIZATION FOR SPECTRAL CLUSTERING

Abstract

Regularization of the classical Laplacian matrices was empirically shown to improve SC *performances in sparse networks. It was further observed that small regularizations seem to be preferable to large ones, but this point was left as a heuristic argument. In this chapter we formally determine a proper regularization which is intimately related to our findings of Chapter 5 that explains why small regularizations should be preferred and that guarantees detectability down to the theoretical threshold.*

Chapter 5 showed that a conveniently parametrization of the Bethe-Hessian matrix allows one to define a *spectral clustering* (SC) algorithm that deals at once with the sparsity and heterogeneity in the degree distribution that typically characterize real-world graphs. Figure 5.5 evidences that the efficiency of our proposed Algorithm 5.1 is unbeaten by all competing spectral algorithms on *degree corrected stochastic block model* (DCSBM)-generated graphs. From this simulation, it however also appears evident that the algorithm of [QR13] obtains very similar performances.

In the literature [QR13; Ami+13; JY13; LR+15; LLV18], regularized spectral clustering has been considered for its ability to improve the concentration properties of random matrices related to graphs with low average degree. A clear representation of the positive effect of regularization is given in Figure 6.1 comparing on a sparse graph the spectrum of $L = D^{-1/2}AD^{-1/2}$ and $L_\tau^{\text{sym}} = D_\tau^{-1/2}AD_\tau^{-1/2}$, where $D_\tau = D + \tau I_n$ on a sparse graph generated from the DCSBM. The aforementioned works, and more specifically

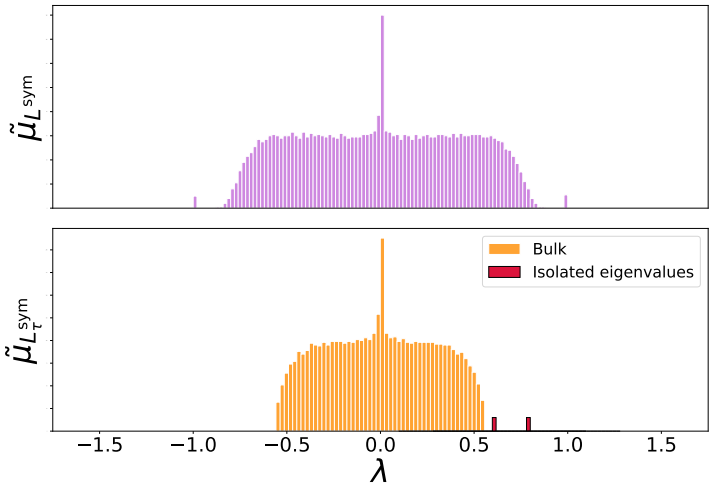


Figure 6.1: **Top**: Spectrum of L^{sym}, **Bottom**: spectrum of L^{sym}_τ with $\tau = c$. For both plots $n = 5\,000$, $\theta_i \sim [\mathcal{U}(3,7)]^3$, $k = 2$, $c = 5$, $c_{\text{out}} = 2$.

[QR13], left some relevant questions unanswered on the role of regularization in SC that we list as follows:

Open questions regarding regularized SC

- The theoretical results of [QR13] together with the proposed choice $\tau = \bar{d}$ (the average degree) are consistent only in the moderately sparse regime in which the expected average degree goes to infinity faster than $\log(n)$. Nevertheless, the proposed algorithm is very efficient also for $c = O_n(1)$, as shown in Figure 5.5.
- There exists no theoretical guarantee or conjecture of whether or not the regularization allows one to achieve detectability down to the theoretical threshold.
- In the literature, the characterization of the parameter τ was never properly addressed and its assignment was left to a heuristic choice. Both in [QR13] and [JY13] the results provided by the authors seem to suggest a large value of τ, but it is observed experimentally that smaller values of τ give better partitions. In the end, the authors in [QR13] settle on $\tau = \bar{d}$ and do not further address the problem of identifying the optimal parametrization.

Working on an explicit relation between the Bethe-Hessian and the regularized Laplacian matrices, in this chapter we provide a natural explanation to regularization. We further give a precise result on the optimal (hardness dependent) regularization as well as on the value of τ needed to achieve detectability down to the theoretical threshold. As it is for Claim 5.1, also in this case the optimal regularization can computed in an unsupervised way. We further naturally explain why small regularizations should be preferred to large ones, answering the three questions raised above. These results create a bridge between statistical physics and regularization techniques.

6.1 MAIN RESULT

As we just mentioned, our main result revolves around a relation between H_r and L_τ^{sym}. Letting $L_\tau^{\text{rw}} = D_\tau^{-1}A$, the mapping unfolds from the following basic remark: for $\zeta_p > 1$ (as defined in Equation (5.3)),

$$[(\zeta_p^2 - 1)I_n + D - \zeta_p A]\boldsymbol{x}_p = 0 \tag{6.1}$$

$$\Leftrightarrow [D + (\zeta_p^2 - 1)I_n]^{-1} A\boldsymbol{x}_p = \frac{1}{\zeta_p}\boldsymbol{x}_p \tag{6.2}$$

$D_\tau = D + \tau I_n$

inducing a natural mapping between the Bethe-Hessian matrix at $r = \zeta_p$ and the regularized random walk Laplacian at $\tau = \zeta_p^2 - 1$. In particular, the vector $\boldsymbol{x}_p$ solution of Equation (6.2) coincides with the eigenvector of $L^{\text{rw}}_{\zeta_p^2-1}$ associated with the eigenvalue $1/\zeta_p$, thoroughly studied in Chapter 5. Note that L_τ^{rw} and L_τ^{sym} have the same eigenvalues and the eigenvectors are trivially related. Our main result states that the regularization $\tau = \zeta_p^2 - 1$ guarantees that the top p eigenvalues of $L_\tau^{\text{rw/sym}}$ are isolated down to the detectability threshold, *i.e.* as soon as $\alpha > \alpha_c$. We formulate this result in the form of a claim.

Claim 6.1. *Consider the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, generated from a k-class DCSBM. Let ζ_p for $2 \le p \le k$ be defined as per Equation (5.3) with $\zeta_1 = 1$ and $\tau \in \mathbb{R}^+$ be such that $\zeta_p^2 - 1 \le \tau \le c\Phi - 1$. Then, under Assumption 3.1 and $c = O_n(1)$, for all large n with high probability, the p largest eigenvalues of the matrix L_τ^{rw} (and equivalently of L_τ^{sym}) are isolated. In particular,*

$$\lambda_p^{\downarrow}(L^{\text{rw}}_{\zeta_p^2-1}) = \frac{1}{\zeta_p}.$$

For the sake of clarity, we recall that Assumption 3.1 is the same under which Claim 5.1 was formulated. Note that Claim 6.1 is not an obvious consequence of the equivalence between Equation (6.1) and Equation (6.2) as it is not clear that the eigenvalue $1/\zeta_p$ of $L^{\text{rw}}_{\zeta_p^2-1}$ corresponds to the p-th largest and that it remains isolated (as is zero in the spectrum of H_{ζ_p}). Figure 6.2 gives a visual representation of the spectrum of $\zeta_p L^{\text{sym}}_{\zeta_p^2-1}$, showing that, indeed, its p-th largest eigenvalue is asymptotically close to one.

The eigenvector $\boldsymbol{x}_p$, associated to the p-th largest isolated eigenvalue of L_τ^{rw} is also the solution to $H_{\zeta_p}\boldsymbol{x}_p = 0$ (from Equation (6.1)) and its properties have been extensively discussed in Chapter 5. Intuitively, by picking τ away from $\zeta_p^2 - 1$, the entries of the p-th eigenvector are likely to be more polluted by the degrees of the network. This suggests that large values of τ are likely to lead to sub-optimal partitions.

Claim 6.1 further asserts that, for any τ in the interval $[\zeta_p^2 - 1, c\Phi - 1]$, the p dominant eigenvalues of L_τ^{rw} are isolated. Since $\zeta_p \le \zeta_k \le \sqrt{c\Phi}$, regardless of the hardness of the detection problem, the k largest eigenvalues

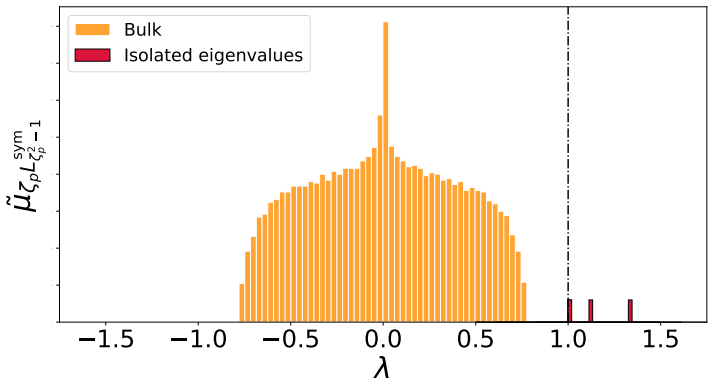


Figure 6.2: Spectrum of $\zeta_p L^{\mathrm{sym}}_{\zeta_p^2-1}$ for $p = 3$. For this simulation, $n = 5\,000$, $k = 3$, $\theta_i \sim [\mathcal{U}(3,10)]^3$, $c = 8$, $\boldsymbol{\pi} = [0.5, 0.2, 0.3]$ and $C_{13} = 1.2, C_{13} = 2.7, C_{32} = 2.4$.

of $L^{\mathrm{rw}}_{c\Phi-1}$ (i.e., $L^{\mathrm{rw}}_{r^2-1}$ for $r = \sqrt{c\Phi}$) must be isolated. The heuristic regularization $\tau = \bar{d}$ proposed in [QR13], while suboptimal according to the claim, is somewhat meaningful as τ must indeed grow with $\bar{d} \approx c$.

As a corollary to Claim 6.1, exploiting Claim 3.3, the matrices $L_\tau^{\mathrm{rw/sym}}$ be exploited to obtain an alternative method to estimate the number of communities, if unknown.

Remark 6.1 (Estimation of the number of classes)**.** *As a direct consequence of Claim 6.1 and Claim 3.3, we have that, for all large n with high probability*

$$k = \left| \left\{ i \,:\, \lambda_i(L_\tau^{\mathrm{rw}})|_{\tau=c\Phi-1} \geq \frac{1}{\sqrt{c\Phi}} \right\} \right|.$$

We now proceed to detail the technical arguments in support of Claim 6.1.

6.2 SUPPORTING ARGUMENTS

To support Claim 6.1, we rely on three intermediary lemmas of which we give formal proof. Our main result is in a form of a claim and not of a theorem because it relies on Claim 5.1 which is not formally proved, but in the following we show that if Claim 5.1 is true, then Claim 6.1 holds as well.

6.2.1 TECHNICAL LEMMAS

Lemma 6.1. *Let D and A be the degree and adjacency matrices of an arbitrary graph of size n. Let $r > 1$ and p an integer between 2 and n.*
If $\lambda_p^{\uparrow}(D - rA) < 0$ then

$$\lambda_p^{\downarrow}\left(L^{\mathrm{rw}}_{-\lambda_p^{\uparrow}(D-rA)}\right) = \frac{1}{r}$$

and both eigenvalues $\lambda_p^{\uparrow}(D - rA)$ and $1/r$ share the same eigenvector.

The fact that the matrix $L^{\mathrm{rw}}_{-\lambda_p^{\uparrow}(D-rA)}$ has an eigenvalue equal to $1/r$ comes easily by construction of $L^{\mathrm{rw}}_{-\lambda_p^{\uparrow}(D-rA)}$. The main result of Lemma 6.1 is to state that this eigenvalue is the p-th largest creating an explicit mapping between the eigenvalues of $D - rA$ (hence of H_r) and L^{rw}_{τ} for a proper value of τ. Note that the condition $\lambda_p^{\uparrow}(D - rA) < 0$ could be loosened, but we do not need a stronger result in the following.

Before proceeding with the proof of Lemma 6.1, for the sake of clarity, we enunciate Courant-Fischer theorem that will be of fundamental use.

Theorem 6.1. [Courant-Fischer, see for instance Bha13] *Let $M \in \mathbb{C}^{n\times n}$ be a Hermitian matrix and U a vector subspace of $\mathbb{C}^n$. Then,*

$$\lambda_p^{\uparrow}(M) = \min_{U:\dim(U)=p} \max_{z\in U, z\neq 0} \frac{z^T M z}{z^T z}$$

$$\lambda_p^{\uparrow}(M) = \max_{U:\dim(U)=n-p+1} \min_{z\in U, z\neq 0} \frac{z^T M z}{z^T z}.$$

We now proceed with the proof of Lemma 6.1.

Proof of Lemma 6.1. Let $r > 1$. For simplicity of notation, let us denote $\lambda_p(r) \equiv \lambda_p^{\uparrow}(D - rA)$. Define the matrix M_r as

$$M_r = -\lambda_p(r) I_n + D - rA = D_{-\lambda_p(r)} - rA$$

where we recall the notation $D_\tau = D + \tau I_n$. Note that one has $\lambda_p^{\uparrow}(M_r) = 0$. Letting $\tilde{M}_r \equiv D^{-1/2}_{-\lambda_p(r)} M_r D^{-1/2}_{-\lambda_p(r)}$, we then have

$$L^{\mathrm{sym}}_{-\lambda_p(r)} = \frac{1}{r}\left[I_n - \tilde{M}_r\right].$$

Recalling that L^{rw}_{τ} and L^{sym}_{τ} share the same spectrum, one has in particular:

$$\lambda_p^{\downarrow}(L^{\mathrm{rw}}_{-\lambda_p(r)}) = \lambda_p^{\downarrow}(L^{\mathrm{sym}}_{-\lambda_p(r)}) = \frac{1}{r}\lambda_p^{\downarrow}\left(I_n - \tilde{M}_r\right) = \frac{1}{r}\left(1 - \lambda_p^{\uparrow}(\tilde{M}_r)\right).$$

Thus, proving the lemma amounts to proving that $\lambda_p^{\uparrow}(\tilde{M}_r) = 0$, which we do now by first proving that $\lambda_p^{\uparrow}(\tilde{M}_r) \leq 0$ and then that $\lambda_p^{\uparrow}(\tilde{M}_r) \geq 0$. Denote by $X = (\boldsymbol{x}_1 | \dots | \boldsymbol{x}_p) \in \mathbb{R}^{n\times p}$ the matrix concatenating the eigenvectors associated to the p smallest eigenvalues of M_r. One has:

$$\forall\, 1 \leq q \leq p,\ \ M_r \boldsymbol{x}_q = \lambda_q^{\uparrow}(M_r)\boldsymbol{x}_q \quad \text{and} \quad \lambda_q^{\uparrow}(M_r) \leq \lambda_p^{\uparrow}(M_r) = 0.$$

Now define the vector space V as $V = \mathrm{span}\left(D^{1/2}_{-\lambda_p(r)} X\right)$. Since $\lambda_p(r)$ is strictly negative by hypothesis, $D_{-\lambda_p(r)} \succ 0$ (is positive definite), which in turn implies that $\dim(V) = p$. By definition of V, $\forall\, \boldsymbol{z} \in V$, $\exists\, \boldsymbol{u} \in \mathbb{R}^p$: $\boldsymbol{z} = D^{1/2}_{-\lambda_p(r)} X \boldsymbol{u}$. From the Courant-Fischer theorem, we can write:

$$\lambda_p^{\uparrow}(\tilde{M}_r) = \min_{U:\dim(U)=p} \max_{z\in U, z\neq 0} \frac{z^T \tilde{M}_r z}{z^T z} \leq \max_{z\in V, z\neq 0} \frac{z^T \tilde{M}_r z}{z^T z}$$

$$\leq \max_{\boldsymbol{u}\in\mathbb{R}^p, \boldsymbol{u}\neq 0} \frac{\boldsymbol{u}^T X^T D^{1/2}_{-\lambda_p(r)} \tilde{M}_r D^{1/2}_{-\lambda_p(r)} X \boldsymbol{u}}{\boldsymbol{u}^T X^T D_{-\lambda_p(r)} X \boldsymbol{u}}$$

i.e.:

$$\lambda_p^{\uparrow}(\tilde{M}_r) \leq \max_{\boldsymbol{u}\in\mathbb{R}^p,\boldsymbol{u}\neq 0} \frac{\boldsymbol{u}^T X^T M_r X \boldsymbol{u}}{\boldsymbol{u}^T X^T D_{-\lambda_p(r)} X \boldsymbol{u}} = \max_{\boldsymbol{u}\in\mathbb{R}^p,\boldsymbol{u}\neq 0} \frac{\sum_{q=1}^{p} (\boldsymbol{x}_q^T \boldsymbol{u})^2 \lambda_q^{\uparrow}(M_r)}{\boldsymbol{u}^T X^T D_{-\lambda_p(r)} X \boldsymbol{u}} \leq 0$$

where for the last step we exploited $\lambda_q^{\uparrow}(M_r) \leq 0$ and $D_{-\lambda_p(r)} \succ 0$. We thus conclude that $\lambda_p^{\uparrow}(\tilde{M}_r) \leq 0$.

To prove that the equality holds, we exploit the second relation of the Courant-Fischer theorem. We define $\bar{X} = (\boldsymbol{x}_p | \dots | \boldsymbol{x}_n) \in \mathbb{R}^{n\times(n-p+1)}$ the matrix concatenating the eigenvectors associated to the $n-p+1$ largest eigenvalues of M_r. For $q \geq p$, $\lambda_q^{\uparrow}(M_r) \geq 0$. We further define $W = \text{span}\left(D_{-\lambda_p(r)}^{1/2} \bar{X}\right)$, satisfying $\dim(W) = n-p+1$. We can write:

$$\lambda_p^{\uparrow}(\tilde{M}_r) = \max_{U:\ \dim(U)=n-p+1} \min_{z\in U, z\neq 0} \frac{\boldsymbol{z}^T \tilde{M}_r \boldsymbol{z}}{\boldsymbol{z}^T \boldsymbol{z}} \geq \min_{z\in W, z\neq 0} \frac{\boldsymbol{z}^T \tilde{M}_r \boldsymbol{z}}{\boldsymbol{z}^T \boldsymbol{z}}$$

i.e.:

$$\lambda_p^{\uparrow}(\tilde{M}_r) \geq \min_{\boldsymbol{u}\in\mathbb{R}^{n-p+1},\boldsymbol{u}\neq 0} \frac{\boldsymbol{u}^T \bar{X}^T M_r \bar{X} \boldsymbol{u}}{\boldsymbol{u}^T \bar{X}^T D_{-\lambda_p(r)} \bar{X} \boldsymbol{u}} = \min_{\boldsymbol{u}\in\mathbb{R}^p,\boldsymbol{u}\neq 0} \frac{\sum_{q=p}^{n} (\boldsymbol{x}_q^T \boldsymbol{u})^2 \lambda_q^{\uparrow}(M_r)}{\boldsymbol{u}^T \bar{X}^T D_{-\lambda_p(r)} \bar{X} \boldsymbol{u}} \geq 0.$$

As a consequence, $\lambda_p^{\uparrow}(\tilde{M}_r) \geq 0$. Combining both inequalities, we obtain that $\lambda_p^{\uparrow}(\tilde{M}_r) = 0$.

The fact that the eigenvectors are shared comes from the following. Let $\boldsymbol{x}_p(r)$ be the eigenvector of $D - rA$ associated to $\lambda_p(r)$, *i.e.*, $[D-rA]\boldsymbol{x}_p(r) = \lambda_p(r)\boldsymbol{x}_p(r)$. This can be re-written as $L^{\text{rw}}_{-\lambda_p(r)} \boldsymbol{x}_p(r) = \frac{1}{r}\boldsymbol{x}_p(r)$.

□

The first Lemma introduces a formal mapping between the p-th smallest eigenvalue of $D - rA$ and the p-th largest eigenvalue of L_τ^{rw} for the choice $\tau = -\lambda_p^{\uparrow}(D-rA)$ Claim 6.1 however gives a result on the spectrum of L_τ^{rw} for $\zeta_p^2 - 1 \leq \tau \leq c\Phi - 1$, hence, unlike Lemma 6.1, it is not limited to a single value of τ. The next lemma takes care of generalizing the result considering the behavior of $\tau = \lambda_p^{\uparrow}(D-rA)$ as a function of r.

Lemma 6.2. *Let D and A be the degree and adjacency matrices of an arbitrary graph. Let p be an integer ≥ 2. Suppose that there exists $r_p > 1$ such that: (i) $\forall\ r \geq r_p$ the eigenvalue $\lambda_p^{\uparrow}(D-rA)$ is simple; (ii) $\lambda_p^{\uparrow}(D-r_pA) < 0$. Then:*

$$\forall\ r \geq r_p, \quad \partial_r \lambda_p^{\uparrow}(D-rA) < 0.$$

In words, once $\lambda_p^{\uparrow}(D-rA)$ becomes negative, it is strictly decreasing.

Proof of Lemma 6.2. Once again, let us write $\lambda_p(r) = \lambda_p^{\uparrow}(D-rA)$ to lighten notations. Let p, r_p be an integer ≥ 2 and a scalar > 1 (if they exist) such

that $\lambda_p(r_p) < 0$. Let $r \geq r_p$. As $\lambda_p(r)$ is an eigenvalue, $\exists\ \boldsymbol{x}_p(r)$ with $||\boldsymbol{x}_p(r)||^2 = 1$ such that:

$$[D - rA]\boldsymbol{x}_p(r) = \lambda_p(r)\boldsymbol{x}_p(r),$$

which implies in particular that $\boldsymbol{x}_p(r)^T A \boldsymbol{x}_p(r) = \frac{1}{r}\boldsymbol{x}_p(r)^T D_{-\lambda_p(r)}\boldsymbol{x}_p(r)$. As we suppose $\lambda_p(r)$ to be simple, we can apply the eigenvalue perturbation theorem [GLO20, for instance]:

$$\partial_r \lambda_p(r) = -\boldsymbol{x}_p^T(r) A \boldsymbol{x}_p(r) = -\frac{1}{r}\boldsymbol{x}_p(r)^T D_{-\lambda_p(r)}\boldsymbol{x}_p(r). \tag{6.3}$$

In Equation (6.3), $D_{-\lambda_p(r_p)} \succ 0$ (as $\lambda_p(r_p) < 0$ by hypothesis), consequently $\partial_r \lambda_p(r)\big|_{r=r_p} < 0$. We now want to show that for all $r > r_p$, $\partial_r \lambda_p(r) < 0$. We proceed with a proof by contradiction.

Suppose that there exists a value $r' > r_p$ such that $\partial_r \lambda_p(r)\big|_{r=r'} \geq 0$. From Equation (6.3) it follows that a necessary (but not sufficient) condition to be verified is that $\lambda_p(r') \geq 0$. From a continuity argument on the function $\lambda_p(r)$, and as $\lambda_p(r_p) < 0$ and $\lambda_p(r') \geq 0$ with $r' > r_p$, there exists $r'' \in (r_p, r')$ such that $\partial_r \lambda_p(r)\big|_{r=r''} > 0$ and $\lambda_p(r'') < 0$. Invoking once again Equation (6.3), no such r'' can exist, invalidating the hypothesis we made by absurd. We thus conclude that

$$\forall\ r \geq r_p, \quad \partial_r \lambda_p(r) < 0, \tag{6.4}$$

finishing the proof.

□

The eigenvalue simplicity assumption of this lemma is technical and enables us to properly define and manipulate the derivative of the p-th smallest eigenvalue. In case of multiplicity, the tools involved are more complicated [GLO20] and not included here. The simplicity assumption is however verified with high probability on the DCSBM-generated graphs considered in Claim 6.1, hence justifying our assumption.

With Lemma 6.2 we established a relation between τ in the regularized Laplacian and r in the Bethe-Hessian, showing that to larger values or r should correspond larger values of τ. We now enunciate our final supporting lemma which states that isolated eigenvalues of H_r are mapped to isolated eigenvalues of L_τ^{rw} while bulk eigenvalues are mapped to bulk eigenvalues.

Lemma 6.3. *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a graph with k classes generated from the sparse DCSBM and let D and A be the corresponding degree and adjacency matrices respectively. Consider an integer p between 2 and k and let ζ_p be defined as per Equation (5.3) and $r \geq \zeta_p$. If $\lambda_p^{\uparrow}(D - rA) < 0$ is isolated with associated eigenvector $\boldsymbol{x}_p$, then $\boldsymbol{x}_p$ is an eigenvector of $L^{\text{rw}}_{-\lambda_p^{\uparrow}(D-rA)}$, whose corresponding eigenvalue is also isolated. Otherwise, for $1 \leq p \leq n$ and $r > 1$, if $\lambda_p^{\uparrow}(D - rA)$ is a bulk eigenvalue of $D - rA$, then the corresponding eigenvalue of $L^{\text{rw}}_{-\lambda_p^{\uparrow}(D-rA)}$ is also a bulk eigenvalue.*

Proof of Lemma 6.3. Consider the values $\{\zeta_p\}_{2\leq p\leq k}$ as defined in Eq.(5.3). As $\mathcal{G}(\mathcal{V},\mathcal{E})$ is generated from a DCSBM we know from Claim 5.1 that these $k-1$ values of ζ_p exist with high probability. Define $\zeta_{k+1} = \sqrt{c\Phi}$ and fix p an integer such that $2 \leq p \leq k$. Let us write once more $\lambda_p(r) = \lambda_p^{\uparrow}(D - rA)$ to lighten notations. Note that $\lambda_p^{\uparrow}(H_{\zeta_p}) = 0, \forall\, 2 \leq p \leq k$ and $\lambda_{k+1}^{\uparrow}(H_{\zeta_{k+1}}) = o_n(1)$, as shown in [SKZ14]. We thus have $\lambda_p(\zeta_p) < 0$, $\forall\, 2 \leq p \leq k+1$ and it is isolated by hypothesis. We can apply Lemma 6.2, for $r_p = \zeta_p$: for all $r \geq \zeta_p$, $\partial_r\lambda_p(r) < 0$ and $\lambda_p(r) < 0$. Consequently, there exists a unique value of $r' \geq r$, satisfying $\lambda_{p+1}(r') = \lambda_p(r)$. From Lemma 6.1, we know that:

- as $\lambda_p(r) < 0$: $\boldsymbol{x}_p(r)$, the eigenvector associated to $\lambda_p(r)$, is also the eigenvector of $L^{\mathrm{rw}}_{-\lambda_p(r)}$ associated to its p-th largest eigenvalue, $1/r$.
- as $\lambda_{p+1}(r') < 0$: $\boldsymbol{x}_{p+1}(r')$, the eigenvector of $D - r'A$ associated to $\lambda_{p+1}(r')$, is also the eigenvector of $L^{\mathrm{rw}}_{-\lambda_{p+1}(r')} = L^{\mathrm{rw}}_{-\lambda_p(r)}$ associated to its $(p+1)$-th largest eigenvalue, $1/r'$.

Thus, the p-th (resp. $(p+1)$-th) largest eigenvalue of $L^{\mathrm{rw}}_{-\lambda_p(r)}$ is $1/r$ (resp. $1/r'$). Consider $r \geq \zeta_p$. By hypothesis, $\lambda_p(r)$ is an isolated eigenvalue, that is, we can write

$$\begin{aligned} O_n(1) &= \lambda_{p+1}(r) - \lambda_p(r) = \lambda_{p+1}(r) - \lambda_{p+1}(r') \\ &= \int_{r'}^{r} dx\; \partial_x\lambda_p(x) = \kappa(r - r') \end{aligned}$$

for some constant $\kappa = O_n(1)$ independent of n, representing the average value of $\partial_x\lambda_p(x)$ on the integration interval. The constant $\kappa = O_n(1)$ because for any $r, r' = O_n(1)$, we have that $\lambda_p(r), \lambda_p(r') = O_n(1)$. The eigengap for $L^{\mathrm{rw}}_{-\lambda_p(r)}$ is $1/r - 1/r' = (r' - r)/(rr')$, thus

$$O_n(1) = O_n\big(\lambda_{p+1}(r) - \lambda_p(r)\big) = O_n(r - r') = O_n\left(\frac{1}{r} - \frac{1}{r'}\right) = O_n(1).$$

So, if $\lambda_p(r)$ is isolated, the eigenvalue $1/r$ of L^{rw}_{τ} is isolated as well.

On the other hand, if $\lambda_p(r)$ is in the bulk, then $\exists\; q\; : |\lambda_p(r) - \lambda_q(r)| = o_n(1)$. By an argument of continuity on $\lambda_p(r)$, one can analogously define an r' satisfying $\lambda_p(r) = \lambda_q(r')$, concluding that in this case $|r - r'| = o_n(1)$ and so eigenvalues in the bulk are mapped into eigenvalues in the bulk. □

With this proof we conclude this technical section in which we enunciated and proved three lemmas that formally relate the eigenvalues of H_r with those of L^{rw}_{τ}. We now proceed with an informal proof of Claim 6.1.

6.2.2 INFORMAL PROOF OF CLAIM 6.1

Based on Lemmas 6.1-6.3 and Claim 5.1 we now proceed to the justification of Claim 6.1. Once again, we adopt the notation $\lambda_p^{\uparrow}(D - rA) = \lambda_p(r)$.

The first part of Claim 6.1 asserts that for $\zeta_p^2 - 1 \leq \tau \leq c\Phi - 1$ the p largest eigenvalues of the matrix L_τ^{rw} are isolated. According to Claim 5.1, for $\zeta_p \leq r \leq \sqrt{c\Phi}$ the eigenvalue $\lambda_p(r)$ is isolated. Thanks to Lemma 6.2, since $\lambda_p(\zeta_p) < 0$ we have: $\partial_r \lambda_p(r) < 0$ for all r in the interval $\zeta_p \leq r \leq \sqrt{c\Phi}$, implying $\lambda_p^{\uparrow}(D - rA) < 0$ for all r in the interval $\zeta_p \leq r \leq \sqrt{c\Phi}$. The hypotheses of Lemma 6.1 and Lemma 6.3 are therefore verified. We can then assert that the eigenvalue $1/r$ is the p-th largest of the matrix $L^{\text{rw}}_{-\lambda_p^{\uparrow}(D-rA)}$ and it is isolated.

Further exploiting Lemma 6.2, letting $\tau(r) = -\lambda_p^{\uparrow}(D - rA)$, we have $\partial_r \tau(r) > 0$ for all r in the interval $\zeta_p \leq r \leq \sqrt{c\Phi}$. The function $\tau(r)$ is thus bijective and increasing on this interval. Since the eigenvalue $\lambda_p^{\uparrow}(D - rA)$ is isolated for all $\zeta_p \leq r \leq \sqrt{c\Phi}$, the corresponding eigenvalue of $L^{\text{rw}}_{\tau(r)}$ equal to $1/r$ is isolated for

$$\zeta_p^2 - 1 = -\lambda_p(\zeta_p) \leq \tau \leq -\lambda_p(\sqrt{c\Phi}).$$

Note now that, for $1 \leq p \leq k$, $\lambda_p^{\uparrow}(D - \sqrt{c\Phi}A) \leq \lambda_{k+1}^{\uparrow}(D - \sqrt{c\Phi}A) = c\Phi - 1$. This implies that for $\zeta_p^2 - 1 \leq \tau \leq c\Phi - 1$, the top p eigenvalues of L_τ^{rw} are certainly isolated.

For the particular case where $r = \zeta_p$, by definition, $-\lambda_p^{\uparrow}(D - \zeta_p A) = \zeta_p^2 - 1 > 0$ and the result is straightforwardly obtained by applying Lemma 6.1, concluding our argument.

With the justification of our main result being settled, we are now in position to close this chapter with some concluding remarks.

6.3 CONCLUSION

In this chapter we showed that there is an explicit relation between the Bethe-Hessian and the regularized Laplacian matrices. Building on this relation, we generalized our Claim 5.1 to L_τ^{sym}. In the end of Chapter 5 three questions concerning SC with the regularized Laplacian matrix were left open. The answers to these questions are the main contribution of this chapter and are summarized as follows.

- In all the works related to t regularization [QR13; Ami+13; JY13; LR+15; LLV18] the approach chosen to study SC consists in providing a bound to $\|M - \mathbb{E}[M]\|$, where M is the matrix under study and $\mathbb{E}[M]$ its expectation. If on the one hand regularization provides good results for moderately dense graphs, on the other hand it is not sufficient to give tight bounds in the regime $c = O_n(1)$. In this chapter we used a completely different method that justifies the role of regularization starting from the inner isolated eigenvalues of non-backtracking matrix, studied in Chapter 5. Our analysis, although not formally rigorous, holds in the sparse regime $c = O_n(1)$.

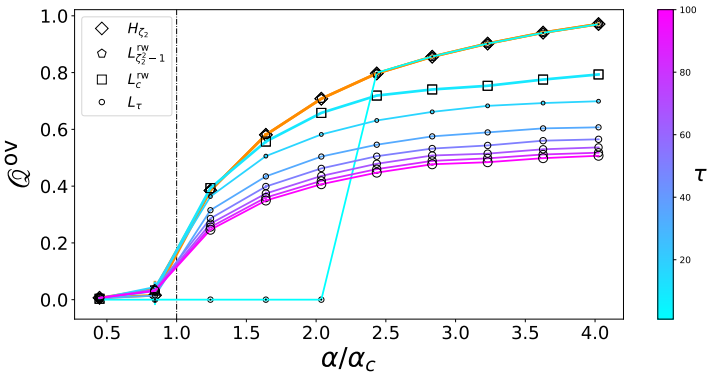


Figure 6.3: Clustering on the eigenvector associated to the second largest eigenvalue of L^{rw}_τ. The circles indicate values of $\tau \in [1, c^2]$: the colour code and the marker size distinguish the values of τ. The squares are obtained for $\tau = c$, while the orange line (with pentagons) is obtained for $\tau = \zeta_2^2 - 1$. Superimposed to this line, the diamonds are the output of Algorithm 5.1. For this simulation $n = 50\,000$, $k = 2$, $c = 10$, $\theta_i \sim [\mathcal{U}(3, 15)]^5$, $\boldsymbol{\pi} = \mathbf{1}_2$. Averages are taken over 5 realizations.

- After considering the sparse regime $c = O_n(1)$, the second natural question concerns whether or not the matrix L^{sym}_τ exhibits isolated eigenvalues down to the detectability threshold and, if so, for what τ. Claim 6.1 gives a precise result stating that indeed the detectability threshold can achieved specifying the interval of values of τ that can accomplish this task.

- An evidence that appeared from the earlier works on regularization is that small values of τ perform better in practice than large ones, even though no clear explanation was provided. We showed that the optimal Laplacian regularization ranges from 0 (for easy problems) to $c\Phi - 1$ for hard ones, naturally justifying the adoption of small regularizations. Figure 6.3 provides further evidence showing for different τ the performance obtained from Algorithm 4.1 setting $M = L^{\text{rw}}_\tau$ and confirming that the proposed choice of τ is optimal.

As a final remark, we recall that also in this case Claim 6.1 can be extended to the setting in which the matrix $C\Pi$ potentially has negative eigenvalues. All comments made in Chapter 5 remain valid also in this case, with the difference that the regularizer $\tau = \zeta_p^2 - 1 > 0$ even if $\zeta_p < 0$. The "disassortative" eigenvalues will be associated to negative eigenvalues of L^{rw}_τ, satisfying $\zeta_p L^{\text{rw}}_{\zeta_p^2-1} = 1$.

Chapter 5, 6 have been devoted to the formulation of two claims under the DCSBM assumption. The resulting Algorithm 5.1 has been introduced as a meta-algorithm for user readability. In the next chapter we consider the problem of how Algorithm 5.1 can be efficiently implemented on real graphs that go beyond the DCSBM assumption. Further considerations will be made on the computational complexity and run time of the resulting algorithm.

7

ALGORITHMIC IMPLEMENTATION

Abstract

This chapter considers how to implement Algorithm 5.1 to obtain an efficient spectral clustering *(SC) method for* community detection *(CD) on* arbitrary *graphs. In particular, as a main contribution we provide an efficient estimator of the number of classes k, based on Claim 3.3 of [SKZ14] and subsequently show how to retrieve the values $\{\zeta_p\}_{p=1,\dots,k}$. The proposed algorithm is then extensively tested on several real-world graphs and its performance is shown to typically outperform the one achieved by the competing state-of-the-art SC techniques.*

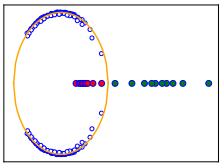
Chapter 5 showed that a sequence of carefully parametrized Bethe-Hessian matrices $\{H_{\zeta_p}\}_{p=1,\dots,k}$ can be exploited to perform SC in sparse graphs with a heterogeneous degree distribution. All the results concerning the values $\{\zeta_p\}_{p=1,\dots,k}$ and the corresponding embedding eigenvectors $\{\boldsymbol{x}_p\}_{p=1,\dots,k}$ are based on the assumption that the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ is a realization of a sparse DCSBM with $n \to \infty$ nodes. Real-world graphs, however, force us to seriously face three main problems. The first is that the setting $n \gg 1$, while it interestingly corresponds to big datasets, it cannot be given for granted. Consequently, an efficient and practical algorithm for CD cannot rely on asymptotic results. Related to this, the second problem lies in the fact that, as we already discussed in Chapter 1, for finite n there is not a clear boundary between the *sparse* and the *dense* settings and our proposed algorithm must be well defined (and efficient) for any average degree. Finally, short cycles are often observed in real social graphs [HL71]: this fact may put in question our

From DCSBM to real-world graphs

theoretical results based on the Bethe approximation, which is appropriate for tree-like graphs.

It has however to be noted that, while Chapter 5 only considers the DCSBM setting to justify Algorithm 5.1, the definition of $\{\zeta_p\}_{p=1,\dots,k}$ ensures their existence for *any* input graph and it does not rely on the $n \gg 1$, nor the sparsity, nor the DCSBM assumptions. In fact, based on several empirical tests, we observed that the following properties of the non-backtracking matrix (Definition 1.8) seem to hold in general:

Spectrum of B for the football *dataset [GN02]*



- All complex eigenvalues come in pairs of complex conjugates and most of them are bounded by a circle on the complex plane of radius $r_{\text{radius}} \approx \sqrt{\lambda_1^{\downarrow|\cdot|}(B)}$.
- The number of real eigenvalues of B, different from $\{-1, 0, 1, \lambda_1^{\downarrow|\cdot|}(B)\}$, is even. Half of the eigenvalues are larger in modulus than r_{radius}, while the other half lies between 1 and r_{radius}. All of them are isolated.

Given the connection between the non-backtracking and the Bethe Hessian matrices, from these two points, we can claim that the steps of Algorithm 5.1 are all well defined on arbitrary graphs. By all means, we can consequently say that the set of ζ_p's are a *property* of the graph for which we have a clear interpretation for the DCSBM. Based on these observations, the goal of this chapter is to design efficient routines to accomplish the steps of Algorithm 5.1 on an arbitrary input graph.

Among the questions that have to be explicitly answered in the following, there is the one concerning the estimation of k, which is necessary to defined the set $\{\zeta_p\}_{p=1,\dots,k}$. Building on the result of [SKZ14], we develop an efficient algorithm (especially for large values of k) to estimate k, which we recall is a typically hard task to accomplish when performing SC.

Our findings result in Algorithm 7.1 which is tested on some real graphs benchmarks comparing its efficiency against the competing spectral methods. The results are reported in Table 7.1.

7.1 IMPLEMENTATION DETAILS

In this section we describe in detail how to efficiently implement the subroutines of our proposed algorithm that we summarize in Algorithm 7.1 which is the implementation adopted in the `CoDeBetHe.jl` package.

Each connected component of the graph is treated independently

Before getting into the details of the steps of Algorithm 7.1, let us recall that a further specificity of real graphs is their possibility to be inherently made of disjoint components. We have instead so far worked with the assumption that $\mathcal{G}(\mathcal{V}, \mathcal{E})$ has a giant component and that the communities are contained within the giant component. In practice, communities may live in disconnected subgraphs (in a two-class DCSBM, this would correspond to

Algorithm 7.1 : Spectral CD in sparse and heterogeneous graphs

Input : Adjancency matrix of undirected, connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$
Output : Estimated number of communities $\hat{k}$, estimated label community vector $\hat{\boldsymbol{\ell}}$.

begin
  $\hat{k} \leftarrow$ `estimate_number_of_classes` (Algorithm 7.2);
  $\boldsymbol{\zeta} \leftarrow$ `compute_`$\boldsymbol{\zeta}$ (Algorithm 7.3);
  Initialize $X \in \mathbb{R}^{n \times \hat{k}}$;
  for $p = 1 : k$ **do**
    $X_{\bullet,p} \leftarrow \boldsymbol{x}_p \in \mathbb{R}^n$, where $\boldsymbol{x}_p$ is the eigenvector with eigenvalue $\lambda_p^{\uparrow}(H_{\zeta_p}) = 0$;
  end
  Normalize the rows of $X_{i,\bullet} \leftarrow X_{i,\bullet} / \|X_{i,\bullet}\|$;
  Estimate community labels $\hat{\boldsymbol{\ell}}$ as output of $\hat{k}$-class *k-means* on the rows of X;
  return $\hat{k}$, $\hat{\boldsymbol{\ell}}$.
end

setting $c_{\text{out}} = 0$). In this case, one can perform a two-stage clustering. First the connected components are detected, looking into the eigenvectors with zero eigenvalue of $H_{\zeta_1} = D - A$. Afterwards, each connected component is treated independently. We will thus consider, without loss of generality, that the real graphs which we will consider are connected.

7.1.1 ESTIMATION OF k

The problem of estimating the number of communities in an unsupervised manner is in general non-trivial. Methods based on the non-backtracking rather than Bethe-Hessian matrix have been studied and exploited to efficiently recover communities regardless of the generative model [LL15].

A possible way to define an estimator of k is as the largest number of negative eigenvalues of H_r as a function of r, or, more succinctly,

$$\hat{k} = \max_{r \geq 1} |i \ : \ \lambda_i(H_r) < 0| \, .$$

This estimator is well defined on arbitrary graphs and has a deep interpretation rooted in statistical physics. However, a direct implementation of this estimator is inefficient because it requires to compute several times the eigenvalues of H_r for different values of r. A crucial result of [SKZ14] is to determine for the SBM (but the result can be easily generalized to the DCSBM) the value of r for which the number of negative eigenvalues of H_r is maximal and is precisely equal to $r = \sqrt{\lambda_1^{\downarrow|\cdot|}(B)} = \sqrt{c\Phi}$. This value of r can be efficiently estimated from the degree sequence, since $c\Phi \approx \sum_{i \in \mathcal{V}} d_i^2 / \sum_{i \in \mathcal{V}} d_i$.

With extensive simulations, we observed however that on general graphs $c\Phi$ is generally not a good approximation for $\lambda_1^{\downarrow|\cdot|}(B)$. For this reason, it is best to compute directly the leading eigenvalue. This can be done efficiently, because all eigenvalues of $B \in \{0,1\}^{|\mathcal{E}_d|\times|\mathcal{E}_d|}$ different from ± 1 are also eigenvalues of the smaller matrix $B' \in \mathbb{R}^{2n\times 2n}$ [Krz+13; CZ20, for instance],

$$B' = \begin{pmatrix} A & I_n - D \\ I_n & 0 \end{pmatrix} \tag{7.1}$$

so all computation involving the eigenvalues of B can be performed on the matrix B'. Computing the p largest eigenvalues of a matrix with m non-zero elements costs $O(mp^2)$ with state-of-the-art methods such as restarted Arnoldi methods [Saa92], hence $\lambda_1^{\downarrow|\cdot|}(B)$ can be efficiently computed in $O(cn)$.

This leads us to the following estimator of k, adopted in the remainder

Estimator of the number of communities

$$\hat{k} = \left| i \,:\, \lambda_i \left(H_{\sqrt{\lambda_1^{\downarrow|\cdot|}(B)}} \right) < 0 \right|.$$

As we showed in Chapter 6, an estimator of k can be designed also on the regularized Laplacian matrix as detailed in the following remark.

Remark 7.1 (Estimating k from L_τ^{sym})**.** *Remark 6.1 states that k can also be estimated using the regularized Laplacian matrix L_τ^{sym} as follows*

$$\hat{k} = \left| i \,:\, \lambda_i \left(L^{\text{sym}}_{\lambda_1^{\downarrow|\cdot|}(B)-1} \right) > \frac{1}{\sqrt{\lambda_1^{\downarrow|\cdot|}(B)}} \right|.$$

The main advantage of this estimator with respect to the one based on H_r is that one needs to compute the eigenvalues with largest rather than smallest algebraic value that, generally, can be done more efficiently.

Moreover, suppose that one wants to include also disassortative communities in the estimation of k. Referring to DCSBM this is the case in which $C\Pi$ may have also negative eigenvalues. The estimators of k for the Bethe-Hessian and the regularized Laplacian can be written, respectively, as

$$\hat{k} = \left| i \,:\, \lambda_i \left(H_{\sqrt{\lambda_1^{\downarrow|\cdot|}(B)}} \right) < 0 \right| + \left| i \,:\, \lambda_i \left(H_{-\sqrt{\lambda_1^{\downarrow|\cdot|}(B)}} \right) < 0 \right|$$

$$\hat{k} = \left| i \,:\, \left| \lambda_i \left(L^{\text{sym}}_{\lambda_1^{\downarrow|\cdot|}(B)-1} \right) \right| > \frac{1}{\sqrt{\lambda_1^{\downarrow|\cdot|}(B)}} \right|,$$

evidencing that L_τ^{sym} allows one to estimate k using a single value of τ, while H_r requires two values of r. In the remainder, the "disassortative contribution"

is however not considered and consequently the two estimators are essentially equivalent. For notation convenience, we will refer to the one based on H_r, but all the results shown below can be generalized to L_τ^{sym}.

Let us now consider the problem of counting the number of negative eigenvalues of $H_{\sqrt{\rho}}$, where $\rho = \lambda_1^{\downarrow|\cdot|}(B)$ for convenience. A naïve implementation simply consists in computing the p smallest eigenvalues $H_{\sqrt{\rho}}$ and check if they are all negative. If the condition is met, then p is increased to $p+1$ and the procedure is iterated until the first positive eigenvalue of $H_{\sqrt{\rho}}$ is found. This implementation is however non efficient, especially for large k, since it requires to compute several times the same eigenvalues. The complexity of this method scales as $O(nc\hat{k}^3)$. A recent workaround strategy [RM15; Tre+16a; Tre+16b] based on *random projections* and *polynomial approximations* allows us to define an efficient routine that initializes p to a value close to $\hat{k}$, which is then computed in a few iterations.

Random projections

Denote with $Y \in \mathbb{R}^{n\times\hat{k}}$ the matrix containing in its columns the eigenvectors associated to the negative eigenvalues of $H_{\sqrt{\rho}}$. Note that, since $H_{\sqrt{\rho}}$ is Hermitian, $Y^TY = I_{\hat{k}}$. Let $\boldsymbol{f} \in \mathbb{R}^n$ be a random vector with zero mean and $\mathbb{E}[\boldsymbol{f}\boldsymbol{f}^T] = I_n$. An estimator of $\hat{k}$ unfolds from the following relations

$$\begin{aligned} k &= \operatorname{Tr}(I_{\hat{k}}) = \operatorname{Tr}(Y^TY) = \operatorname{Tr}(YY^T) = \operatorname{Tr}\big(YY^T \underbrace{\mathbb{E}[\boldsymbol{f}\boldsymbol{f}^T]}_{I_n}\big) \\ &= \mathbb{E}\left[\operatorname{Tr}(YY^T\boldsymbol{f}\boldsymbol{f}^T)\right] = \mathbb{E}\big[\operatorname{Tr}(\underbrace{\boldsymbol{f}^TYY^T\boldsymbol{f}}_{\in\mathbb{R}})\big] = \mathbb{E}\left[\boldsymbol{f}^TYY^T\boldsymbol{f}\right] \\ &= \mathbb{E}\big[\boldsymbol{f}^TY\underbrace{Y^TY}_{I_{\hat{k}}}Y^T\boldsymbol{f}\big] = \mathbb{E}\left[\|YY^T\boldsymbol{f}\|^2\right]. \end{aligned} \tag{7.2}$$

Equation (7.2) shows that $\hat{k}$ can be estimated taking the expectation of the square norm of $YY^T\boldsymbol{f}$. At this stage, this procedure is however pointless, since the matrix Y (whose size is determined by $\hat{k}$) needs to be computed. We now proceed showing that an approximate method can be devised to estimate $\hat{k}$ without directly computing Y.

Polynomial approximation

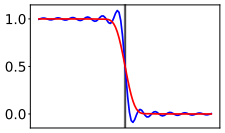
Let us introduce the door function $\Theta_{ab}(x) = \mathbb{I}_{x\in[a,b]}$. The number of negative eigenvalues of $H_{\sqrt{\rho}}$ is equal to the number of eigenvalues of $H'_{\sqrt{\rho}} = H_{\sqrt{\rho}} - \lambda_1^{\uparrow}(H_{\sqrt{\rho}})I_n$ in the interval $[0, -\lambda_1^{\uparrow}(H_{\sqrt{\rho}})]$. Letting $\Lambda'_{\sqrt{\rho}} \in \mathbb{R}^{n\times n}$ be the diagonal matrix containing the eigenvalues of $H'_{\sqrt{\rho}}$, we denote with $\big(\Theta_{ab}(\Lambda'_{\sqrt{\rho}})\big)_{ij} = \Theta_{ab}\left(\big(\Lambda'_{\sqrt{\rho}}\big)_{ij}\right)$. Denoting for simplicity $\lambda_1^{\uparrow}(H_{\sqrt{\rho}}) = \lambda_{\min}$, the following relation holds

$$YY^T = Y\,\Theta_{0,-\lambda_{\min}}(\Lambda'_{\sqrt{\rho}})\,Y^T \equiv \Theta_{0,-\lambda_{\min}}(H'_{\sqrt{\rho}}).$$

Obtaining the expression of $\Theta_{0,-\lambda_{\min}}(H_r)$ is still unpractical because all eigenvalues must be computed. An approximation of $YY^T \approx \tilde{Y}\tilde{Y}^T$ can however be efficiently obtained replacing $\Theta_{0,-\lambda_{\min}}(x)$ with its polynomial approximation[1] $\tilde{\Theta}_{0,-\lambda_{\min}}(x)$ of order Q. Denoting with $c_{0\leq q\leq Q}$ the coefficients of the polynomial, in fact

$$\tilde{Y}\tilde{Y}^T = \tilde{\Theta}_{0,-\lambda_{\min}}(H'_{\sqrt{\rho}}) = \sum_{q=0}^{Q} c_q \left(H'_{\sqrt{\rho}}\right)^q ,$$

Chebychev (blue) and Jackson-Checychev (red) approximations of the step function



then, exploiting the fact that $H_{\sqrt{\rho}}$ and $H'_{\sqrt{\rho}}$ share the same eigenvectors, $YY^T \approx \tilde{Y}\tilde{Y}^T$. This allows us to define an estimator of $\hat{k}$ as follows[2]

$$\hat{k} \approx \frac{1}{N}\sum_{i=1}^{N}\left\|\sum_{q=0}^{Q} c_q \left(H'_{\sqrt{\rho}}\right)^q \boldsymbol{f}_i\right\|^2 , \tag{7.3}$$

where N denotes the number of random vectors generated to compute the empirical mean. This estimator of $\hat{k}$ is obtained in $O(cnNQ)$ operations. Due to the good concentration properties of their norm, if $\boldsymbol{f}_i$ are Gaussian random vectors, a very small value of N (*e.g.* $N = 4$) is sufficient to obtain a good estimator of the expectation. On the other hand, the value of Q must be carefully chosen to guarantee a good approximation of the step function. In practice, it is convenient to increase Q until the variation in the estimation of $\hat{k}$ remains below a given threshold. To give an order of magnitude, Q is, for the typical problems it was tested on, of order of 100. The fact that the estimator of $\hat{k}$ based on random projection has a $\hat{k}$-independent complexity, makes it particularly suited for large values of $\hat{k}$.

The strategy to efficiently compute the number of negative eigenvalues is then to give it a first rough approximation as per Equation (7.3) and then proceed with the naïve method detailed before to compute this quantity exactly. Empirically, it is convenient to compute an initial estimate of $\hat{k}$ for $\hat{k} \approx 5$ or greater. For smaller values, the naïve direct computation might be more efficient, but both algorithms provide a fast solution in this case.

The results so far presented allow us to detail Algorithm 7.2 to efficiently estimate the number of communities on an arbitrary graph. For simplicity, we indicate here Q as an input of the algorithm. Neglecting the dependence

1 The choice of which polynomial approximation to consider is not straightforward. One possible choice is to use Chebychev polynomials as they have a guarantee on the infinite norm of the approximation error. However, they tend to create Gibbs oscillation around sharp cutoffs of the function to approximate. As the function we wish to approximate here is a step function, it is customary to choose Jackson-Chebychev polynomials which explicitly dampen these unwanted oscillations [Jay+99].

2 Note that, while $\hat{k} \in \mathbb{N}$, the estimator in Equation (7.3) is defined in $\mathbb{R}$.

Algorithm 7.2 : `estimate_number_of_classes`

Input : Adjancency matrix of undirected connected graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, Q: order of the polynomial approximation, N: number of random vectors

Output : Estimated number of commnities $\hat{k}$

begin
- $\rho \leftarrow \lambda_1^{\downarrow|\cdot|}(B')$, with B' as in Equation (7.1);
- $H'_{\sqrt{\rho}} \leftarrow H_{\sqrt{\rho}} - \lambda_1^{\uparrow}(H_{\sqrt{\rho}})I_n$;
- $\{c_q\}_{0\le q\le Q} \leftarrow$ Jackson-Chebishev coefficients of the polynomial approximation of $\Theta_{0,-\lambda_1^{\uparrow}(H'_{\sqrt{\rho}})}$;
- Generate N Gaussian random vectors with zero mean and idendity covariance matrix $\boldsymbol{f}_{1\le i\le N} \in \mathbb{R}^n$;
- $k_{\text{init}} = \texttt{ceil}\left(\frac{1}{N}\sum_{i=1}^{N}\left\|\sum_{q=0}^{Q} c_q \left(H'_{\sqrt{\rho}}\right)^q \boldsymbol{f}_i\right\|^2\right)$;
- Compute $\lambda_i^{\uparrow}(H_{\sqrt{\rho}})$ for $1 \le i \le k_{\text{init}}$;
- $k \leftarrow \arg\max_i \left(\lambda_i(H_{\sqrt{\rho}}) < 0\right)$ for $1 \le i \le k_{\text{init}}$;
- **while** $\lambda_k^{\uparrow}(H_{\sqrt{\rho}}) < 0$ **do**
  - $k \leftarrow k + 1$;
  - Compute $\lambda_k^{\uparrow}(H_{\sqrt{\rho}}) < 0$;
- **end**
- **return** $k - 1$;

end

on Q and N, the complexity of Algorithm 7.2 scales in the worst case scenario in which k_{init} is very far from $\hat{k}$ as $O(nc\hat{k}^3)$. However, if k_{init} is close to $\hat{k}$, Algorithm 7.2 converges in $O(nc\hat{k}^2)$ operations. To give a practical measure of computational times we tested Algorithm 7.2 on a laptop[3] for different values of k on graphs generated from the *stochastic block model* (SBM) with $n = 10^5$, $c = 8$, $c_{\text{out}} = 4/k$. The value of $\hat{k}$ was computed for $k = 2, 8, 30$ in $8, 8, 10$ seconds.

We now move to the analysis of the second subroutine, dedicated to estimate the values of ζ_p.

7.1.2 ESTIMATION OF $\{\zeta_p\}$

A crucial step of Algorithm 7.1 is to compute the vector $\boldsymbol{\zeta} \in \mathbb{R}^{\hat{k}}$, containing the optimal values of r in which the eigenvalues of H_r should be computed. In the remainder, for simplicity, we will denote $k = \hat{k}$. From Claim 5.1, a simple relation holds between the values ζ_p and the isolated eigenvalues of B for the DCSBM, in fact

3 The laptop's RAM is 7.7Gb with Intel Core i7-6600U CPU @ 2.6GHz x 4.

$$\zeta_p = \frac{c}{\lambda_p^{\downarrow}(C\Pi)} + o_n(1) = \frac{\lambda_1^{\downarrow|\cdot|}(B)}{\lambda_p^{\downarrow|\cdot|}(B)} + o_n(1). \tag{7.4}$$

This estimation of ζ_p via B', defined in Equation (7.1) is computationally efficient but we observed that it can be quite inaccurate for graphs not generated from the DCSBM. Conversely, a naïve line search for $\zeta_p \in (1, \sqrt{\rho})$ (where $\rho = \lambda_1^{\downarrow|\cdot|}(B)$) satisfying $\lambda_p^{\uparrow}(H_{\zeta_p}) = 0$ is computationally inefficient.

We propose here a faster method, motivated by a Courant-Fischer theorem argument (Theorem 6.1). In a few words, starting from an initial guess r_0, we devise an iterative sequence $r_0, r_1, \dots$, such that $r_t \to \zeta_p$. The convergence is guaranteed when setting $r_0 = \zeta_{p+1}$, under the convention $\zeta_{k+1} = \sqrt{\rho}$. The values of ζ_p are then estimated from the largest, ζ_k, to the smallest, ζ_2. Let us now get to the mathematical details of this subroutine. We denote by $X_{r_t}^T = (\boldsymbol{x}_1^T(r_t), \dots, \boldsymbol{x}_p^T(r_t))^T \in \mathbb{R}^{n\times p}$ where $\boldsymbol{x}_p(r_t)$ is the eigenvector of H_{r_t} corresponding to the p-th smallest eigenvalue, $\lambda_p^{\uparrow}(H_{r_t})$, while $\Lambda_{r_t} = \operatorname{diag}\left(\lambda_1^{\uparrow}(H_{r_t}), \dots, \lambda_p^{\uparrow}(H_{r_t})\right)$. For another value $r' \neq r_t$, and applying the Courant-Fischer theorem, we can write

$$\begin{aligned}\lambda_p^{\uparrow}(H_{r'}) &= \min_{U:\dim(U)=p} \max_{z\in U, z\neq 0} \frac{\boldsymbol{z}^T H_{r'} \boldsymbol{z}}{\boldsymbol{z}^T \boldsymbol{z}} \\ &\leq \max_{\boldsymbol{u}\in\mathbb{R}^p} \frac{\boldsymbol{u}^T X_{r_t}^T H_{r'} X_{r_t} \boldsymbol{u}}{\boldsymbol{u}^T \boldsymbol{u}} = \lambda_1^{\downarrow}(X_{r_t}^T H_{r'} X_{r_t})\end{aligned}$$

i.e.:

$$\begin{aligned}\lambda_p^{\uparrow}(H_{r'}) &\leq \lambda_1^{\downarrow}(X_{r_t}^T[(r'^2 - {r_t}^2)I_n - (r' - r_t)A + H_{r_t}]X_{r_t}) \\ &\leq (r'^2 - {r_t}^2) + \lambda_1^{\downarrow}(\Lambda_{r_t} - (r' - r_t)X_{r_t}^T A X_{r_t}).\end{aligned}$$

We can further simplify the earlier expression by exploiting the identity:

$$\Lambda_{r_t} = X_{r_t}^T H_{r_t} X_{r_t} = ({r_t}^2 - 1)I_p + X_{r_t}^T D X_{r_t} - r_t X_{r_t}^T A X_{r_t}.$$

We thus obtain

$$\lambda_p^{\uparrow}(H_{r'}) \leq \frac{1}{r_t} \underbrace{\left[(r' - r_t)(1 + r'r_t) + \lambda_1^{\downarrow}\big((r_t - r')X_{r_t}^T D X_{r_t} + r'\Lambda_{r_t}\big)\right]}_{f_{r_t}(r')}. \tag{7.5}$$

We now study the function $f_{r_t}(r')$ for $r_t \in (\zeta_p, \sqrt{\rho})$ and define $r_{t+1} \in (\zeta_p, r_t)$ as the solution (if it exists) to

$$f_{r_t}(r_{t+1}) = 0. \tag{7.6}$$

The idea is to iteratively approach ζ_p from the right and substitute $r_t \leftarrow r_{t+1}$. If $r_{t+1} \in (\zeta_p, r_t)$ as defined above exists, then $\{r_t\}_{t\geq 0}$ is a lower-bounded decreasing sequence: it thus converges to a limit r_∞ (potentially different from

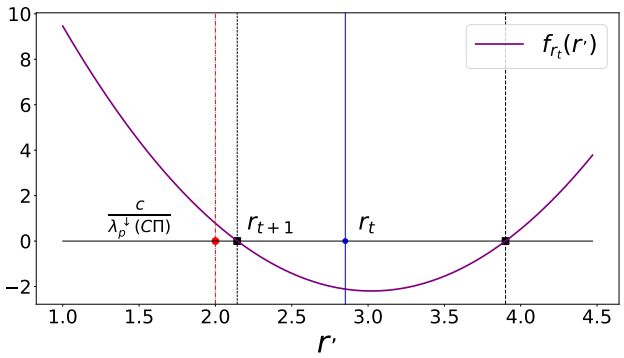


Figure 7.1: Plot of the $f_{r_t}(r')$ (in purple). The blue continuous line indicates the value of r_t, the red dashed dotted line the theoretical value of $\zeta_p = c/\lambda_p^{\downarrow}(C\Pi)$ for the DCSBM. The two dashed black lines are the two roots of $f_{r_t}(r')$, the smaller of which is r_{t+1}. For this simulation $n = 50\,000$, $k = 2$, $c = 5$, $c_{\text{out}} = 2.5$, $r = \sqrt{c\Phi}$, $\theta_i \sim [\mathcal{U}(3,10)]^3$, $\boldsymbol{\pi} \propto \mathbf{1}_k$.

ζ_p). Exploiting [RB69, Theorem in I.4], denoting with F, G two Hermitian matrices and γ a scalar, $\lambda_1^{\downarrow}(F + \gamma G)$ is a convex function of γ. As a consequence, the function $f_{r_t}(r')$ is convex and has either no root or two roots. Since $\lim_{|r'|\to\infty} f_{r_t}(r') = +\infty$, it is enough to find a value of r' for which $f_{r_t}(r') < 0$ to prove that this function has two roots. With a straightforward computation, one can verify that $f_{r_t}(r_t) < 0$, so $f_{r_t}(r')$ has two roots, satisfying $r_{t+1} < r_t$ and $r^+ > r_t$. By construction, since $\lambda_p^{\uparrow}(H_{r'})$ is negative in the considered interval and $\lambda_p^{\uparrow}(H_{r'}) \leq f_{r_t}(r')/r_t$, if $f_{r_t}(r_{t+1}) = 0$, then $\zeta_p \leq r_{t+1}$. Consequently, $f_{r_t}(r_{t+1}) = 0$ has a unique solution satisfying

$$\zeta_p \leq r_{t+1} < r_t$$

and the algorithm converges to $r_\infty = \lim_{t\to\infty} r_t \geq \zeta_p$. We are left to prove that $r_\infty = \zeta_p$. By convergence of r_t, we have $r_{t+1} - r_t = o_t(1)$. Plugging this relation solution into Equations (7.5,7.6), we obtain

$$\lambda_1^{\downarrow}(r_{t+1}\Lambda_{r_{t+1}}) = r_{t+1}\lambda_p^{\uparrow}(H_{r_{t+1}}) = o_t(1).$$

Since $\lambda_p^{\uparrow}(H_r) = 0$ has a unique solution ($r = \zeta_p$) in the interval $r \in (1, \sqrt{\rho})$, we obtain $r_t = \zeta_p + o_t(1)$ and so

$$r_\infty = \zeta_p.$$

The initial value of r can be chosen as $r_0 = \zeta_{p+1}$ (setting $\zeta_{k+1} = \sqrt{\rho}$) that certainly falls in the right interval for all the ζ_p.

Summarizing, the algorithm builds on two parts: (i) the computation of a $p \times p$ matrix obtained from the eigenvectors of H_{r_t} with computational cost of $O(ncp^2)$, (ii) a subsequent line-search using this matrix, with computational cost $O(p^2)$. The advantage of this method is that the line search (which requires many iterations) is computationally cheap, while the most expensive part of the algorithm needs to be performed much fewer times

Algorithm 7.3 : `compute_ζ`

Input : adjacency matrix of a connected, undirected graph $\mathcal{G}$,
number of classes k

Output : Vector $\boldsymbol{\zeta} = \{\zeta_1, \ldots, \zeta_k\}$

begin
  $r \leftarrow \sqrt{\lambda_1^{\downarrow|\cdot|}(B')}$, where B' is defined in Equation (7.1) ;
  $p \leftarrow k$;
  while $p > 1$ **do**
    repeat
      Compute $\Lambda_r = \mathrm{diag}\left(\lambda_1^{\uparrow}(H_r), \ldots, \lambda_p^{\uparrow}(H_r)\right)$ and
       $X_r \in \mathbb{R}^{n\times p}$, with $H_r X_r = X_r \Lambda_r$;
      $\Gamma_r \leftarrow X_r^T D X_r \in \mathbb{R}^{p\times p}$;
      $r^* \leftarrow r' : \lambda_1^{\downarrow}(r'\Lambda_r + (r-r')\Gamma_r) = (r-r')(1+rr')$,
       with line-search on $r' \in (1, r)$;
      $r \leftarrow r^*$
    until *convergence*;
    $\delta \leftarrow$ multiplicity of $\lambda_p^{\uparrow}(H_r) = 0$;
    Set $\zeta_p, \zeta_{p-1}, \cdots, \zeta_{p-\delta+1}$ to r;
    $p \leftarrow p - \delta$;
  end
  return $\boldsymbol{\zeta} = \{\zeta_1, \ldots, \zeta_k\}$;
end

with respect to the greedy line-search to obtain the same accuracy. The total complexity of the algorithm needed to compute the vector $\boldsymbol{\zeta} = (\zeta_1, \ldots, \zeta_k)^T$, scales as $O(nck^3)$.

Algorithm 7.3 summarizes our proposed routine to obtain the vector $\boldsymbol{\zeta}$. Note that, although the subroutine `compute_ζ` only outputs the vector $\boldsymbol{\zeta}$, it can also be used to directly compute the informative eigenvectors $\{\boldsymbol{x}_p\}$. Figure 7.1 provides a typical iteration of Algorithm 7.3, while Figure 7.2 shows the output of the computation of $\boldsymbol{\zeta}$ and confirms the accuracy of the proposed algorithm. On the top line the algorithm is tested on a network created from the DCSBM, for which $\lambda_1^{\downarrow|\cdot|}(B)/\lambda_p^{\downarrow|\cdot|}(B)$ is a valid estimator for ζ_p. The horizontal line indicates $\sqrt{\rho} = \sqrt{c\Phi}$ which is the upper bound of ζ_p. In our simulations, the affinity matrix $C \in \mathbb{R}^{k\times k}$ is generated randomly (see the caption of Figure 7.2), for the first two points we obtained $\lambda_3^{\downarrow}(C\Pi) < \sqrt{c\Phi}$, invalidating Assumption 3.1. In this case we see that $c/\lambda_p^{\downarrow}(C\Pi) > \sqrt{c\Phi}$ and the corresponding estimated value of ζ_p saturates at $\sqrt{c\Phi}$. On the contrary, whenever Assumption 3.1 is verified, the estimate of ζ_p is correct.

In the bottom line we compare the two methods to estimate ζ_p on the three real networks that clearly show that these two methods are different on graphs not generated from the DCSBM. Given that Subroutine 7.3 provides a direct computation of values of ζ_p, it should be preferred.

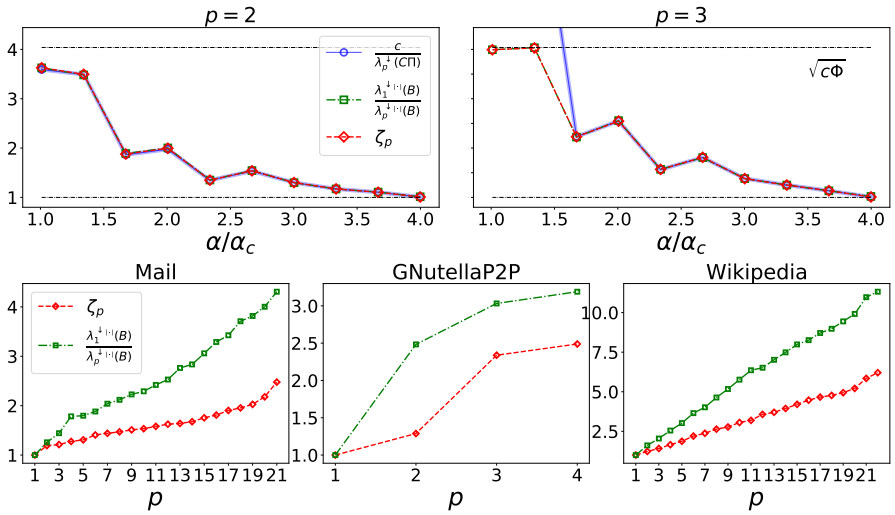


Figure 7.2: (**Top**: DCSBM setting) Exact value of $c/\lambda_p^{\downarrow}(C\Pi)$ ($p = 2$ on the left and $p = 3$ on the right) in blue, compared to the estimate of ζ_p obtained using the dominant eigenvalues of B (green) and its direct computation using Algorithm 7.3 (red), for problems of different hardness. For this simulation, $n = 50\,000$, $k = 3$, $\theta_i \sim [\mathcal{U}(3,10)]^3$, $c = 10$. For each point $\pi_i \sim \mathcal{N}(1/k, 1/2k)$ and the off-diagonal elements of C are distributed as $\sim \mathcal{N}(c_{\text{out}}, c_{\text{out}}/k)$ and $c_{\text{out}} : 0.1 \to 7.5$. (**Bottom**: real networks) Estimate of the values of ζ_p as function of p, computed as $\lambda_1^{\downarrow|\cdot|}(B)/\lambda_p^{\downarrow|\cdot|}(B)$ (green squares) vs the direct computation of ζ_p as the output of Subroutine 7.3 (red diamonds) on the three real networks taken from [LK14]. The value of $\hat{k}$ is estimated according to Algorithm 7.2.

For illustrative purposes, we tested the execution time of Algorithm 7.3 for a *stochastic block model* (SBM) network with $k = 4$ classes of equal size, $n = 10^5$, $c_{\text{in}} = 26.20$ on the diagonal elements of C and $c_{\text{out}} = 4.60$ for the off diagonal elements. The values of ζ_p were computed to machine precision in approximately 5 seconds on a laptop using our `CoDeBetHe.jl` Julia implementation. The complexity of this subroutine scales linearly with n, and thus can be applied to large networks but cubically with respect to k (and not quadratically as usual in the spectral clustering context) decreasing the computational efficiency when a rather large number of classes is present.

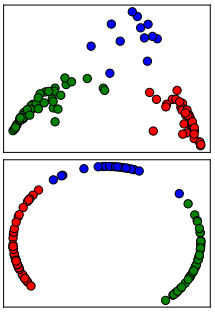
We now proceed with the last technical step of Algorithm 7.1, which concerns the projection of the embedded points on a unitary hypersphere.

7.1.3 PROJECTION OF THE EMBEDDED POINTS ON A HYPER-SPHERE

The study performed so far identifies the presence of k informative eigenvalues and describes the content of the associated k eigenvectors $X \equiv [\boldsymbol{x}_1, \dots, \boldsymbol{x}_k] \in \mathbb{R}^{n \times k}$ with $\boldsymbol{x}_p$ the eigenvector associated to the p-th smallest eigenvalue of H_{ζ_p} (in particular, the vectors $\boldsymbol{x}_p$'s need not be orthogonal). The rows

$X_{1,\bullet}, \dots, X_{n,\bullet} \in \mathbb{R}^k$ of the matrix X form a k-dimensional feature for every node, which are used in a last small-dimensional clustering step, usually employing the *k-means* algorithm. The fact that *k-means* is particularly efficient when the low-dimensional clusters are quite "isotropic" strongly motivates the need for the entries of the vectors $\boldsymbol{x}_p$ not to be affected by the node degrees (which would otherwise spread the clusters unevenly).

The embedding obtained by Algorithm 7.1 on the graph polbooks *with (bottom) and without (top) the projection step*



Yet, to further tackle residual degree dependence, a classical method, prior to *k-means*, consists in normalizing all vectors $X_{i,\bullet}$ to $\|X_{i,\bullet}\| = 1$ (this is Step 4 of Algorithm 4.1) . This method is motivated by the assumption that the degree dependence in each X_{ij} is separable from the label dependence, a fact that is verified in sufficiently dense DCSBM networks [Jin+15] and to some extent also in sparser graphs [QR13]. Besides, under this normalization, the *k-means* algorithm is restricted to the unitary hypersphere, improving its convergence to the genuine solution, especially for large values of k.

As such, while our proposed algorithm naturally discards degree dependence in the entries of X under the DCSBM setting, the reality of practical networks may disrupt this expected behavior and the projection of the vectors $X_{i,\bullet}$ on the unit hyper-sphere both alleviates this deleterious effect and further improves the convergence of *k-means*. We thus adopt this normalization step in our final Algorithm 7.1 and will confirm its practical gains when clustering real graphs.

7.1.3.1 *Computational complexity*

The total theoretical complexity of Algorithm 7.1 is dominated by Algorithm 7.2 and 7.3 that both run in $O(nck^3)$, as the *k-means* step costs only $O(nk^2)$. This is to compare with the usual complexity of spectral clustering algorithms on sparse graphs that are in $O(nck^2)$ [see for instance TL20]. This additional cost comes with a better classification performance in many real-world graphs, as discussed in the next section. To give an order of magnitude of computation times, running Algorithm 7.1 on a SBM with $n = 10^5$ (resp., 10^6), $k = 4$, $c_{\text{in}} = 26.20$ on the diagonal elements of C and $c_{\text{out}} = 4.60$ on the off diagonal terms takes approximately 15 (resp., 450) seconds if k is not known and 6 (resp., 100) seconds if k is known a priori.

7.2 NUMERICAL RESULTS ON REAL GRAPHS

In this section we compare the performance of Algorithm 7.1 versus competing spectral methods on real-world networks [LK14; Zac77; Lus+03; GN02; AG05]. Table 7.1 shows the performance of different SC algorithms on real-world networks. Measuring the quality of an inferred partition $\hat{\boldsymbol{\ell}}$ is in general not straightforward since communities are not uniquely defined. We propose two different scores to accomplish this task. One is the modularity[4]

4 Note that the measure of the modularity is meaningful on *assortative* or *disassortative* networks but not on "hybrid" networks for which a more involved description would be needed.

[NG04], $\mathscr{Q}_A^{\mathrm{Mod}}(\hat{\boldsymbol{\ell}})$, defined in Equation (4.3), where we recall that high values of $\mathscr{Q}_A^{\mathrm{Mod}}$ correspond to good quality partitions. Alternatively, the partition quality is evaluated in terms of (normalized) posterior negative log-likelihood of the DCSBM, $\mathscr{Q}_A^{\mathrm{likelihood}}$:

$$\mathscr{Q}_A^{\mathrm{likelihood}}(\hat{\boldsymbol{\ell}}) = -\frac{1}{2|\mathcal{E}|}\left[\sum_{(ij)\in\mathcal{E}} \log\left(\hat{\theta}_i\hat{\theta}_j\frac{\hat{C}_{\hat{\ell}_i,\hat{\ell}_j}}{n}\right) + \sum_{(ij)\notin\mathcal{E}} \log\left(1-\hat{\theta}_i\hat{\theta}_j\frac{\hat{C}_{\hat{\ell}_i,\hat{\ell}_j}}{n}\right)\right],$$

where $\hat{\theta}_i = d_i/\bar{d}$ and $\hat{C}_{ab} = \left(\sum_{i\,:\,\hat{\ell}_i=a}\sum_{j\,:\,\hat{\ell}_j=b} A_{ij}\right) / \left(\sum_{i\,:\,\hat{\ell}_i=a}\sum_{j\,:\,\hat{\ell}_j=b}\theta_i\theta_j\right)$. Good quality clustering correspond, in this case, to low values $\mathscr{Q}_A^{\mathrm{likelihood}}$.

The results are compared for different clustering algorithms on 15 real-world networks of increasing size. For all networks, the number of communities, when not available, is estimated with Algorithm 7.2 and then the same value is used for all competing techniques (which in general do not provide their own dedicated estimator of k). The underlined numbers in the k column indicate instead that k is known. Furthermore, for all networks, community detection is performed only on the largest connected component of the graph and n, c, Φ, k refer to the characteristics of this dominant connected component. Given the embedding X, 5 iterations of *k-means* are run and the best partition $\hat{\boldsymbol{\ell}}$ (in terms of *k-means* is kept) and the scores $\mathscr{Q}_A^{\mathrm{Mod}}, \mathscr{Q}_A^{\mathrm{likelihood}}$ are computed. To keep the stochastic nature of *k-means* into account, this step is repeated for 8 times and Table 7.1 reports the results of $\mathscr{Q}_A^{\mathrm{Mod}}, \mathscr{Q}_A^{\mathrm{likelihood}}$ in terms of `mean` $\pm$ `standard deviation`.

The first score column of Table 7.1 indicates the output of the meta-algorithm Algorithm 5.1 for which ζ_p is estimated from $\lambda_1^{\downarrow|\cdot|}(B)/\lambda_p^{\downarrow|\cdot|}(B)$. The second score column provides the output of Algorithm 7.1 in which the last step of projection on the hypersphere is not performed. Comparing these two columns, it is clear that Algorithm 7.1 generally provides much better partitions both in terms of $\mathscr{Q}_A^{\mathrm{Mod}}$ and $\mathscr{Q}_A^{\mathrm{likelihood}}$, as a consequence of $\lambda_1^{\downarrow|\cdot|}(B)/\lambda_p^{\downarrow|\cdot|}(B)$ being an inappropriate estimator for ζ_p in general. The following columns display the results scores obtained by Algorithm 7.1 with the projection step (highlighted in cyan), by clustering based on the leading eigenvectors of A [LR+15], by the Bethe-Hessian as per [SKZ14], by the non-backtracking matrix as per [Krz+13], by the random walk Laplacian as per [SM00] and by the symmetric normalized Laplacian of [QR13].

The algorithm of [SM00] based on L^{rw} provides in certain cases very competitive partitions (*e.g.*, in the Wikipedia dataset) but is quite unreliable as it may dramatically fail in others (GNutella P2P and Polblogs). Algorithm 7.1 without the normalization step provides systematically good partitions, all comparable to those of L_τ^{sym}. This is an evidence that Algorithm 7.1 effectively produces a node embedding which is significantly resilient to degree heterogeneity. Finally, the improved version of Algorithm 7.1 including the projection step further improves the quality of the partition on most datasets, providing on all datasets but Wikipedia the highest reported modularity and lowest, or second lowest measured value of $\mathscr{Q}_A^{\mathrm{likelihood}}$.

Dataset	n	c	Φ	k	Alg 4.1$_B$	Alg 7.1$_{\text{wp}}$	Alg 7.1	A	$H_{\sqrt{c\Phi}}$	B	L^{rw}	L_τ^{sym}
Karate	34	4.6	1.7	2	**0.86** **0.37**	**0.86** **0.37**	**0.86** **0.37**	**0.86** **0.37**	**0.86** **0.37**	**0.86** **0.37**	0.98 0.36	**0.86** **0.37**
Dolphins	62	5	1.3	2	**1.25** **0.38**	**1.25** **0.38**	**1.25** **0.38**	1.38 ±0.01 0.22 ±0.01	1.30 ±0.03 0.32 ±0.03	1.38 0.22	**1.25** **0.38**	**1.25** **0.38**
Polbooks	105	8.4	1.4	3	**1.17** **0.51**	1.16 0.50	**1.17** **0.51**	1.26 ±0.01 0.46	1.21 0.50	1.28 ±0.01 0.47 ±0.01	**1.16** 0.50	1.22 **0.51**
Football	115	10.7	1	12	**0.83** **0.60**	**0.83** **0.60**	**0.83** **0.60**	**0.83** **0.60**	**0.83** **0.60**	**0.83** **0.60**	**0.83** **0.60**	**0.83** **0.60**
Mail	1133	9.6	1.9	21	1.95 ±0.01 0.45 ±0.01	**1.89 ±0.02** 0.49 ±0.01	**1.88 ±0.01** **0.52 ±0.01**	2.13 ±0.01 0.31	2.03 ±0.01 0.40	2.05 ±0.01 0.36 ±0.01	1.99 ±0.03 0.50 ±0.02	**1.87 ±0.01** **0.51**
Polblogs	1222	27.4	3	2	**1.50** **0.43**	**1.50** **0.43**	**1.50** **0.43**	1.62 0.25	1.63 0.27	1.65 0.24	1.73 0.00	**1.50** **0.43**
Tv	3892	8.9	3	41	2.01 ±0.03 0.55 ±0.02	1.89 ±0.09 0.70 ±0.06	**1.50 ±0.01** **0.84**	1.99 ±0.09 0.53 ±0.04	2.02 ±0.05 0.53 ±0.02	2.05 ±0.05 0.53 ±0.02	2.06 ±0.15 0.59 ±0.12	**1.47 ±0.01** 0.79 ±0.01
Facebook	4039	43.7	2.4	55	1.17 ±0.04 0.52 ±0.01	1.05 ±0.06 **0.76 ±0.02**	**0.88 ±0.01** **0.76**	1.13 ±0.03 0.44 ±0.03	1.21 ±0.03 0.47 ±0.01	1.21 ±0.03 0.47 ±0.02	1.07 ±0.14 **0.75 ±0.05**	**0.87 ±0.01** 0.59 ±0.01
GrQc	4158	6.5	2.8	29	2.40 ±0.04 0.50 ±0.02	2.22 ±0.11 0.68 ±0.09	**1.89 ±0.01** **0.80**	2.39 ±0.06 0.49 ±0.04	2.40 ±0.04 0.50 ±0.02	2.39 ±0.01 0.50 ±0.01	2.44 ±0.09 0.58 ±0.10	**1.90 ±0.02** 0.78
Power grid	4941	2.7	1.5	25	3.63 ±0.02 0.32 ±0.01	2.61 ±0.01 0.92	**2.57 ±0.01** **0.93**	3.84 ±0.01 0.17 ±0.01	3.63 ±0.03 0.35 ±0.03	3.71 ±0.01 0.28 ±0.01	2.61 ±0.01 **0.92**	2.80 ±0.02 0.86 ±0.01
Politicians	5908	14.1	3	62	1.78 ±0.04 0.52 ±0.02	1.60 ±0.05 **0.83 ±0.01**	1.43 ±0.01 **0.84 ±0.01**	1.94 ±0.06 0.54 ±0.02	1.90 ±0.02 0.50 ±0.02	1.92 ±0.02 0.47 ±0.01	1.77 ±0.11 0.78 ±0.07	**1.40 ±0.01** 0.73 ±0.01
GNutella P2P	6299	6.6	2.7	4	3.32 ±0.01 0.20 ±0.01	3.28 0.26	**3.21** **0.40**	3.33 0.19±0.01	3.37 0.14	3.38±0.01 0.13±0.01	3.41 0.00	3.27 0.35
Wikipedia	7066	28.3	5.1	22	1.93 0.20	**1.88 ±0.01** 0.23 ±0.01	**1.88** 0.26	1.99±0.01 0.15	1.97 0.17	1.96 0.14±0.01	1.94 ±0.03 **0.38 ±0.02**	**1.88** 0.27
HepPh	11204	21.0	6.2	60	1.75 ±0.06 0.39 ±0.02	1.98 ±0.05 0.37 ±0.06	1.52 ±0.01 **0.57**	1.74 ±0.08 0.42 ±0.04	1.79 ±0.04 0.39 ±0.02	1.68 ±0.05 0.44 ±0.03	2.12 ±0.07 0.25 ±0.09	**1.44 ±0.01** 0.51 ±0.01
Vip	11565	11.6	4.4	53	2.60 ±0.03 0.34 ±0.01	2.48 ±0.13 **0.58 ±0.15**	2.22 ±0.01 **0.60 ±0.01**	2.72 ±0.02 0.27 ±0.01	2.64 ±0.03 0.32 ±0.01	2.67 ±0.03 0.30 ±0.01	2.65 ±0.01 0.50 ±0.12	**2.18 ±0.01** 0.54 ±0.01

Table 7.1: Log-likelihood and modularity comparison on real networks [LK14; Zac77; Lus+03; GN02; AG05]. When the standard deviation is below 0.01 is not indicated.

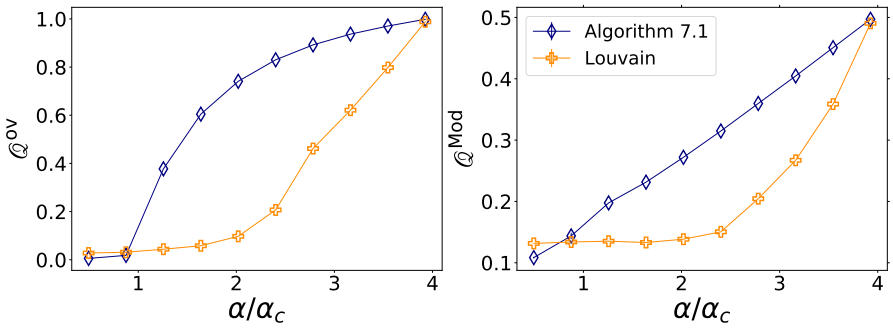


Figure 7.3: Comparison of the overlap (left) and modularity (right) resulting from the label estimation of Algorithm 7.1 (blue diamonds) and Louvain algorithm (orange crosses) applied on the giant component of $\mathcal{G}$, as function of the detection hardness. For this simulation, $n = 50\,000$, $k = 2$, $\pi \propto \mathbf{1}_2$, $c = 8$. Averages are taken over 10 samples.

To test Algorithm 7.1 we computed the vector ζ up to machine error precision, that is, the convergence stopping criterion in Algorithm 7.3 is met when r's update is below machine precision. Notably, due to the non-linearity of the *k-means* step, larger errors in the estimate of the values of ζ lead, within a certain range, to the same partition. The same classification precision can therefore be reached in fewer iterations, thus faster, if needed.

Let us mention that, comparing the columns "Alg 7.1_{wp}" and "$H_{\sqrt{c\Phi}}$", it appears evident that Algorithm 7.1 indeed achieves good quality embeddings on real-world graphs. In addition, it appeared from our simulations that the projection step on the unitary hypersphere of the rows of X significantly helps (both in terms of quality of the partitioning and of the variance of the score) the convergence of *k-means* at a low computational cost.

As a side comment, although formally not strictly comparable on even grounds, we evaluated the performances reached by Algorithm 7.1 against the popular greedy Louvain method [Blo+08] using its *scikit-network* implementation [Ped+11]. The Louvain method comes with an estimate of the number of communities and relies on a different notion of communities than the one we used. Specifically, it is a hierarchical algorithm which looks for a partition maximizing the modularity, while Algorithm 7.1 relies on the DCSBM assumption. Figure 7.3 shows that Algorithm 7.1 largely outperforms the Louvain method on a DCSBM graphs, both in terms of overlap (left) and modularity (right), except below α_c. There, the Louvain method reaches higher modularity, but this is likely an incidental artifact of the modularity optimization constraint [GSPA04].

The situation of real networks is less straightforward. We indeed observed that when the number of communities estimated by both algorithms is similar, they both produce node partitions with a similar modularity. This is however no longer the case when the estimated k are more distinct (*e.g.*, on GNutella P2P). While we recommend our estimation method of k due to its interpretability (that we discussed all along the chapter), we acknowledge that the Louvain algorithm often provides very competitive outcomes, at a

smaller computational cost but at the expense of any theoretical guarantee. Similar considerations can be made in relation to the more recently proposed Leiden method [TWVE19].

7.3 CONCLUSION

This chapter considered the problem of providing an efficient and practical implementation to our proposed Algorithm 5.1, itself inspired from Claim 5.1. We kept the focus on two main problems: on the one hand defining fast routines to accomplish the steps on Algorithm 5.1, on the other hand to design an algorithm which does not heavily rely on the DCSBM assumption and that can perform CD on arbitrary graphs.

In particular, one of the main contributions is the design of Algorithm 7.2, needed to estimate k. We discussed at length in the previous chapters that determining the number of communities on a graph in an unsupervised way is, in general, a very challenging task. Algorithm 7.2 is directly inspired from the results of [SKZ14], but what is truly novel is its efficient implementation.

The result of our analysis leads to Algorithm 7.1 which has been tested on several real-world graphs, confirming that its ability to perform CD on arbitrary graphs. Given the high clustering performance achieved by Algorithm 7.1, as we can also claim that the intuitions encoded in Claim 5.1 for the DCSBM are, to some extent, also verified on arbitrary graphs.

The results reported in Table 7.1 and Figures 5.5, 5.6 put us in position to take a step back and look at our results from a different angle. We can observe that, while the results of Algorithm 7.1 and [QR13] are always the best, the other competing spectral algorithm lead to very good or very poor partitions, according to the setting considered. In the next chapter we show that a unified framework can be devised to interpret the aforementioned SC algorithms, correctly predicting when and why each algorithm fails or succeeds at finding a high quality node partition.

8

A UNIFIED FRAMEWORK FOR SPECTRAL CLUSTERING

Abstract

This chapter provides a critical analysis and interpretation of the competing spectral clustering *(SC) algorithms detailed in the former chapters. It will in particular be shown that it is possible to predict and explain under a unified framework when the aforementioned algorithms are expected to yield good or poor performances. These results further promote Algorithm 7.1 as a powerful candidate for* spectral clustering *(SC) over the competing methods.*

This chapter serves as a conclusion to Part II which studied SC for *community detection* (CD), proposing a novel algorithm. The two main contributions considered so far concern a theoretical analysis of Algorithm 5.1 on *degree corrected stochastic block model* (DCSBM)-generated graphs and a careful implementation capable of performing CD on arbitrary graphs that do not verify the DCSBM assumption, leading to Algorithm 7.1. We here close our analysis of SC for CD with a high level look over the problem, based on the results obtained and detailed in the previous chapters.

The remainder of the chapter is structured as follows. First we consider the algorithms of [LR+15; Jin+15; New06]which are based on the use of the eigenvectors of the adjacency (or of the closely related modularity) matrix. Similarly to the Bethe-Hessian matrix, we argue that a mapping with statistical physics is capable to motivate also these algorithms and naturally justifies the superiority of Bethe-Hessian-based approach.

Laplacian-based algorithms are then considered, explaining how Algorithm 7.1 provides an "optimum" between the classical Laplacian matrices [Fie73; SM00; NJW01b] and the regularized ones [SKZ14; QR13].

These results allow us to claim that Algorithm 7.1 solves the crucial problem of the ambiguous *definition* of sparsity, being adapted to operate in both the dense and the sparse regimes.

8.1 ADJACENCY-BASED ALGORITHMS

In this section we relate Algorithm 7.1 with the SC algorithms of [LR+15; Jin+15] that, referring to Algorithm 4.1, impose the choice $M = A$ and [Newo6], which instead uses a rank one perturbation of the adjacency matrix, $M = A - \frac{dd^T}{2|\mathcal{E}|}$, showing that these algorithms unfold naturally following the procedure used to derive the Bethe-Hessian matrix, but for a different variational approximation.

Recall from Chapter 2 that the Bethe approximation for the Ising model is a variational approximation that is particularly suited when dealing with sparse graphs. A simpler, alternative variational method is represented by the *naïve mean field* (NMF) approximation, also known in the inference literature as *naïve Bayes* [Ris+01].

Let us consider an Ising-like Hamiltonian without external fields (Equation (2.2)) defined on the graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$:

The Ising Hamiltonian

$$\mathcal{H}(\boldsymbol{s}) = -\sum_{(ij)\in\mathcal{E}} J_{ij} s_i s_j.$$

We recall from Chapter 2 that the two fundamental steps of variational approximation are: (i) defining a parametric free energy

The variational free energy

$$\tilde{F}_\beta(\boldsymbol{q}) = \sum_{\boldsymbol{s}} p_{\boldsymbol{q}}(\boldsymbol{s})(\beta\mathcal{H}(\boldsymbol{s}) + \log p_{\boldsymbol{q}}(\boldsymbol{s})),$$

where $p_{\boldsymbol{q}}(\boldsymbol{s})$ is a probability distribution whose expression defines the variational approximation and (ii) find the set of parameters $\boldsymbol{q}$ minimizing $\tilde{F}_\beta(\boldsymbol{q})$.

In the Bethe approximation, $p_{\boldsymbol{q}}(\boldsymbol{s})$ is the expression of the Boltzmann distribution (2.1) on trees, making it adapted to sparse graphs. The NMF approximation is instead obtained assuming all spins to be independent, *i.e.*

The NMF approximation

$$p^{\mathrm{MF}}_{\hat{\boldsymbol{m}}}(\boldsymbol{s}) = \prod_{i\in\mathcal{V}} \frac{1 + \hat{m}_i s_i}{2}. \tag{8.1}$$

The distribution $p^{\mathrm{MF}}_{\hat{\boldsymbol{m}}}$ simply factorizes over all nodes and $\hat{m}_i$ is the expectation of s_i according to Equation (8.1). We then can write the corresponding variational free energy as follows:

$$\tilde{F}^{\mathrm{MF}}_\beta(\hat{\boldsymbol{m}}) = -\sum_{(ij)\in\mathcal{E}} \beta J_{ij}\hat{m}_i\hat{m}_j + \sum_{i\in\mathcal{V}}\sum_{s_i} \frac{1+\hat{m}_i s_i}{2} \log\left(\frac{1+\hat{m}_i s_i}{2}\right).$$

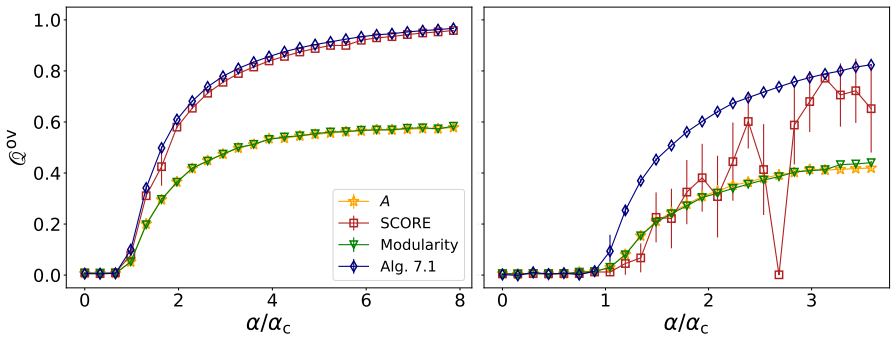


Figure 8.1: Overlap comparison as a function of the hardness of the detection of DCSBM-generated graphs with $n = 10\,000$, $c = 200$, $k = 2$, $\theta_i \sim [\mathcal{U}(3,10)]^4$. Yellow stars correspond to SC as per [LR+15], red squares as per [Jin+15], green triangles as per [Newo6] and the blue diamonds are the result of Algorithm 7.1. **Left**: dense regime, $c = 200$. **Right**: sparse regime, $c = 8$. Averages are taken over 10 samples.

Following the same procedure as in Chapter 2 to obtain the Bethe-Hessian matrix, we now derive the Hessian matrix of $\tilde{F}_\beta^{\mathrm{MF}}(\hat{\boldsymbol{m}})$ at the *paramagnetic* point $\hat{\boldsymbol{m}} = \mathbf{0}_n$, which we denote with H_β^{MF}.

The NMF Hessian at the paramagnetic point

$$\left(H_\beta^{\mathrm{MF}}\right)_{ij} = \left.\frac{\partial^2 \tilde{F}_\beta^{\mathrm{MF}}(\hat{\boldsymbol{m}})}{\partial_i \hat{m}_i \partial \hat{m}_j}\right|_{\hat{\boldsymbol{m}}=\mathbf{0}_n} = (I_n - \beta J)_{ij}\,.$$

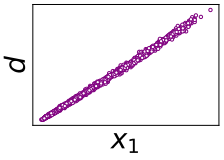
Also in this case, the smallest eigenvalues of H_β^{MF} and their corresponding eigenvectors bring information on the local minima of the free energy and, in the context of CD, can be exploited to determine the community structure of $\mathcal{G}(\mathcal{V},\mathcal{E})$. The smallest eigenvalues of H_β^{MF} are however trivially related to the largest eigenvalues of J and the two matrices share the same eigenvectors. Unlike the Bethe-Hessian matrix, however, in this case the temperature β plays no role in determining the eigenvectors of H_β^{MF} and, consequently, it is pointless to tune it to an optimal value.

The relation with SC comes naturally into play when imposing $J = A$ that corresponds precisely to the choice made to derive the Bethe-Hessian matrix H_r. This choice naturally leads to a NMF version of SC based on the eigenvectors of H_β^{MF} associated to its smallest eigenvalues, the smallest of which corresponds to the ferromagnetic configuration (it is the Perron-Frobenius eigenvalue of A), while the subsequent eigenvectors are correlated with the community structure. From these elements, the relation with [LR+15; Jin+15] unfolds naturally.

If instead one chooses $J = A - \frac{\boldsymbol{d}\boldsymbol{d}^T}{2|\mathcal{E}|}$, the NMF approximation leads to the algorithm of [Newo6] (at least in the case of $k = 2$ communities) in which the *ferromagnetic* configuration is energetically penalized and clustering should be performed on the eigenvector associated to the smallest (resp. largest) eigenvalue of M_β^{MF} (resp. $A - \frac{\boldsymbol{d}\boldsymbol{d}^T}{2|\mathcal{E}|}$). It has to be noted however that (i) unlike the adjacency matrix, the modularity matrix is dense, hence the eigenvectors are computed in a less efficient way; (ii) on sufficiently dense

graphs, the Perron-Frobenius eigenvector of A is strongly aligned with the degree vector.[1] Given that A is Hermitian, all its eigenvectors (expect the leading one) are essentially orthogonal to $\boldsymbol{d}$ and are, therefore, a close approximation of the eigenvectors of $A - \frac{\boldsymbol{d}\boldsymbol{d}^T}{2|\mathcal{E}|}$, as well. For these reasons the algorithm of [Newo6] and of [LR+15] are expected to have a very similar performance (as confirmed by Figure 8.1), with the first being computationally faster. The *SCORE* algorithm of [Jin+15] is then an efficient improvement of [LR+15] which allows one to keep he degree heterogeneity into account. The performances of [Newo6; LR+15; LR+15] are compared with the output of Algorithm 7.1 in Figure 8.1 for $k = 2$ class DCSBM graphs with high average degree on the left and with low average degree on the right.

Scatter plot of the leading eigenvector of A ($\boldsymbol{x}_1$) vs $\boldsymbol{d}$ for a dense DCSBM graph



With these results at hand, we can affirm that the algorithms of [Newo6; LR+15; Jin+15] can be directly compared with Algorithm 7.1 since they can be obtained following the same procedure, yet for a different variational approximation. The final step is to claim that the Bethe approximation is overall more accurate[2] than NMF [MWJ13] on random graphs. Intuitively, this can be understood saying that NMF can be seen as a *first order* approximation (only the first moment of $\boldsymbol{s}$ is kept into account), while the Bethe approximation is a *second order* approximation (that considers also the spins correlations).

The NMF approximation is appropriate on dense but not on sparse graphs

More specifically, the NMF is appropriate for dense graphs [OS01], being in fact asymptotically exact on the fully connected *Curie-Weiss* model [KPW13] in which $J = J\mathbf{1}_n\mathbf{1}_n^T$. It performs instead poorly on graphs with a low average degree, as a consequence of the spectral behavior of A in the sparse regime. On the opposite, the Bethe approximation (hence H_r) is adapted to deal with sparse graphs. One may wonder, however, if it is still a good option to study dense graphs. It can be shown [OS01] that in the dense regime, the Bethe free energy can be approximated with the so-called *TAP* free energy [TAP77] which differs from the NMF free energy due to existence of a correction, called *Onsager reaction term* [Ons36].

The appropriateness of the Bethe approximation on dense random graphs can further be understood from a *random matrix theory* (RMT) perspective. We showed in fact in Section 3.2.2 that there exists a trivial relation between the spectral properties of the non-backtracking and adjacency matrices on dense SBM-generated graphs and further observed empirically that this relation holds also in the DCSBM case.

For this reason, we claim that SC based on the Bethe-Hessian matrix H_r is expected[3] to provide better performances than SC based of the adjacency

1 This is formally proved for the DCSBM in [Jin+15], but intuitively it holds in general as a consequence of $\boldsymbol{d}$

2 Recall that the fundamental task of a variational approach is to approximate the free energy (Definition 2.1). This problem is well defined, independently of CD and for any graph.

3 The claim of a class of SC algorithms being uniformly superior to another one is, certainly, very hard to motivate, also recalling the difficulty to assess the performance of CD on practical tasks. The word "expected" plays here the fundamental role to underline that the assertion has not to be considered as a strict rule (counterexamples in which SC based on A performs better than H_r may be potentially found) but rather as a general trend.

matrix A. Moreover, unlike A, the matrix H_r seems to be a suitable candidate to perform SC for *any* average degree, solving the crucial problem introduced by the ambiguity of the definition of sparsity. For this reason, H_r allows one to bridge the dense and sparse regimes.

8.2 LAPLACIAN-BASED ALGORITHMS

This section draws some connections between the Laplacian-based algorithms of [Fie73; SM00; NJW01b; QR13], including also the Bethe-Hessian of [SKZ14]. We here refer to SC based on $L^{\mathrm{sym/rw}}$ and $L_\tau^{\mathrm{sym/rw}}$ under the same terms since from a spectral viewpoint, these pairs of matrices are closely related. Referring to the "classical" normalized Laplacian matrices, the only difference between [SM00] and [NJW01b] resides in step 4 of Algorithm 4.1. Similar comments can be made also for [QR13].

Let us first consider the graph Laplacian matrix $L = D - A$ of [Fie73]. The Bethe-Hessian matrix can be considered as a *regularized* version of L. The central result of Claim 5.1 revolves around the exploitation of a series of Bethe-Hessian matrices $\{H_{\zeta_p}\}_{p=1,\dots,k}$ whose expression is reported in Equation (8.2) for convenience

$$H_r = (r^2 - 1)I_n + D - rA. \tag{8.2}$$

For the DCSBM we claimed that, under proper hypotheses, the optimal regularization to extract information from the p-th eigenvector of H_r is $r = \zeta_p = \frac{\lambda_1^{\downarrow}(C\Pi)}{\lambda_p^{\downarrow}(C\Pi)} + o_n(1)$, where we recall that $(C\Pi)_{ab}$ is the average number of neighbours in class b of a node in class a. For easy detection problems, the off-diagonal terms of $C\Pi$ simply go to zero and, due to Assumption 5.1 according to which $C\Pi\mathbf{1}_k = c\mathbf{1}_k$, the eigenvalues of $C\Pi$ become degenerate, hence, for all $1 \leq p \leq k$, $\zeta_p \to 1$ and $H_{\zeta_p} \to D - A$. For this reason, we can interpret the *combinatorial graph Laplacian* matrix as the trivial clustering limit of the optimal Bethe-Hessian matrix. We further recall from Remark 4.3 that the multiplicity of the zero eigenvalue of $L = H_{r=1}$ equals the number of connected components of a given graph. Identifying communities with the connected components corresponds to the extreme case in which the number of connections between nodes in the same community is larger than the connections across communities (which is equal to zero) and is the simplest setting in which CD can be performed. From the definition of $\zeta_p^{(j)}$ in Equation (5.2), the choice $r = 1$ unfolds naturally to detect the connected components of a graph.

The combinatorial graph Laplacian matrix is the trivial clustering limit of the optimal Bethe-Hessian

From a statistical physics viewpoint, the choice $r = 1$ (irrespective of the actual value of ζ_p) corresponds to the *zero temperature limit*[4] in which the free energy $F = \beta U - S \to \beta U$ neglects the contribution of the entropy. For this reason, when ζ_p is far from 1, imposing $r = 1$ (*i.e.* , using the graph

4 Recall that β is the inverse temperature and that $\beta \mathrm{J} = \mathrm{ath}(r^{-1})$.

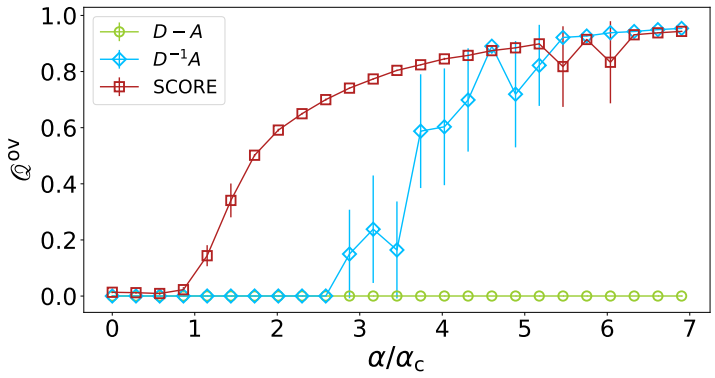


Figure 8.2: Overlap comparison as a function of the hardness of the detection of DCSBM-generated graphs with $n = 10\,000$, $c = 200$, $k = 2$, $\theta_i \sim [\mathcal{U}(3,10)]^4$. Green dots correspond to SC as per [Fie73], light-blue diamonds as per [SMoo] and red dots as per [Jin+15]. Averages are taken over 10 samples.

Laplacian matrix) is equivalent to assuming a negligible contribution from the noise in $\mathcal{G}(\mathcal{V},\mathcal{E})$ and hence leads to over-fitting. In fact, the typical problem of SC based on L is that single nodes are isolated into one community, due to the existence of uninformative local minima in the energy spectrum.

Disassortative and bipartite graphs

Similar remarks can be made concerning the setting of disassortative communities. As it was commented in Section 5.4, in this case $\zeta_p < 0$ and the trivial limit corresponds to $\zeta_p = -1$ for which $H_{\zeta_p} = D + A$, also known as *signless Laplacian* [CRSo7]. It is known that this matrix is a useful tool to study *bipartite* graphs that correspond the "extreme disassortative setting" in which connections exist only between nodes in different classes.

The existing sparse SC algorithms are a "worst case" limit of the optimal Bethe-Hessian

If on the one hand $D - A$ corresponds to the trivial clustering limit, the Bethe-Hessian $H_{\sqrt{c\Phi}}$ of [SKZ14] is the opposite regime, considering a *worst-case scenario*. In fact, the condition $\zeta_p = \sqrt{c\Phi}$ corresponds (for k classes of equal size) to the *detectability threshold*, below which it is conjectured to be impossible to recover communities with an efficient algorithm [Dec+11]. The choice $r = \sqrt{c\Phi}$ allowed the authors of [SKZ14] to design an algorithm capable of making non-trivial clustering as soon as theoretically possible but that is generally suboptimal, as extensively commented in the previous chapters. Figure 5.6 clearly summarizes the behavior of the regularizer r: small values of r lead to better partitions, provided that they guarantee the existence of isolated informative eigenvalues in the spectrum of H_r.

Let us now briefly consider instead the *regularized Laplacian matrix*

$$L_\tau^{\text{sym}} = D_\tau^{-1/2} A D_\tau^{-1/2},$$

where $D_\tau = D + \tau I_n$. In Chapter 6 we showed that the optimal regularization consists in the choice $\tau = \zeta_p^2 - 1$. In the trivial regime, this leads to $\tau \to 0$, for which $L_\tau^{\text{sym}} \to L^{\text{sym}}$ (a similar relation holds of course for L_τ^{rw} and L^{rw}). Very similar comments to the ones made for $D - A$ can be done

also for the *normalized Laplacian matrices* that, indeed, are obtained, like L, from the relaxation of an optimization problem. Similarly to [SKZ14], the regularized Laplacian of [QR13] considers instead an "almost" worst case scenario. In order to guarantee detectability down to the threshold with L_τ^{sym} as in [SKZ14], one must pick $\tau = c\Phi - 1$, while [QR13] proposes heuristically $\tau = c$ which is close to the parametrization which guarantees a non-trivial result also in the hardest regime.

Take home messages

With these observations being laid out, we conclude this section with two fundamental take-home messages. The Laplacian matrices (with the exception of the Bethe-Hessian of [SKZ14]) have been typically studied on *dense* graphs because the mathematical tools to address this regime are generally more powerful and better known than those in the sparse regime. Unlike the matrix A that is derived from a variational approximation that is unsuited for the sparse regime, however, the poor performances of the Laplacian matrices are rather a consequence of an overly optimistic choice of the parametrization (r or τ) which corresponds to assuming the clustering problem to have a trivial solution. To confirm this claim, Figure 8.2 compares the overlaps a function of $\alpha = (c - c_{\text{out}})/\sqrt{c}$, obtained by the algorithms of [Fie73; SM00; LR+15] for *dense* DCSBM graphs with $k = 2$ communities of equal size. This plot evidences that the algorithmic threshold of [LR+15] (based of A) achieves smaller values of α than the one of [Fie73; SM00].

Concluding, while Section 8.1 argues that H_r is a good candidate to bridge the dense and the sparse regimes, Section 8.2 exhibits that Algorithm 7.1 finds the optimal between the trivial clustering regime (assumed *de facto* by [Fie73; SM00; NJW01b]) and the hard one considered in [SKZ14; QR13].

8.3 CONCLUSION

In this chapter we provided a critical overview of the main methods for SC for CD taken under consideration in this manuscript and showed how they can all be understood in a simple unified framework that fosters the adoption of Algorithm 7.1 for CD. The aforementioned methods were mostly developed in parallel by different communities and motivated with different lines of arguments. The existence of a unified view (yet, we acknowledge, somewhat biased by the statistical physics perspective) is an important contribution to relate and interpret the existing results. Notably, we discovered that the hyper-parameters involved in these algorithms must be smartly selected, this selection being a function of the difficulty of the clustering problem: finding these hyper-parameters is key to the clustering performance.

The necessity to adapt the choice of the graph representation matrix to the hardness of the problem at hand is a profound observation, likely not restricted to CD. In light of this observation, it may appear that some common optimization problems, beyond clustering, have similarly been devised in the

past to perform well in easy scenarios, but would require a task-hardness related update to perform better in harder settings.

In the next chapters, we provide practical examples of how the results of Part II can be generalized to more involved scenarios. In particular, in Chapter 9 we consider the problem of dynamical CD that realistically keeps into account for the fact that real-world systems modelled by networks typically evolve with time. Moreover, Chapter 10 treats our extension to the case of weighted graphs, with the specific application to the sparsification of kernel matrices used to obtain cost efficient data clustering spectral algorithms.

GENERALIZATIONS

This part of the manuscript presents the extensions of our works on spectral clustering for community detection, presented in Part II. First, in Chapter 9 we consider the setting of dynamical community detection on graphs generated from the *dynamical degree corrected stochastic block model*, proposing a novel spectral algorithm. Then, in Chapter 10 we treat the problem of i.i.d. sparsification of kernel matrices for spectral clustering.

9

SPECTRAL CLUSTERING IN DYNAMICAL GRAPHS

Abstract

This chapter extends the results of Part II considering the problem of community detection *(CD) in dynamical graphs in which the community structure evolves over time. The dynamical setting is of utmost importance in modern network science, since most systems modelled by graphs indeed have a non-trivial dependence on time. Keeping the graph dynamical component into account allows us to efficiently exploit information coming from different times, consequently improving the clustering quality. As a main result, we develop a novel spectral algorithm for dynamic CD, studied on graphs generated from the* dynamical degree corrected stochastic block model *(DDCSBM).*

In Part II the problem of *community detection* (CD) with *spectral clustering* (SC) techniques has been deeply investigated on *static* graphs. The word "static" here simply refers to the absence of the notion of time. Taking a step back and referring to Chapter 1, we must recall that graphs have a scientific interest for their ability to represent and model real world systems of interacting agents that are rarely static. As a consequence, graph representations

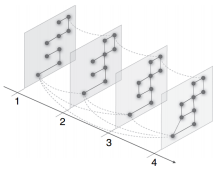
are typically snapshots or averages of the system's interactions occurred in a given time interval.

To give a practical example, consider the *karate* network [Zac77] representing the friendships between 34 members of a karate club. The most significant interactions have been established studying the club members relationships in the a period ranging from 1970 to 1972: the resulting graph representation encodes therefore an average of the members' interactions over the considered time window.

Like in the case of the *karate* network, most systems that can be modelled with graphs have an intimate dynamical component and a richer description can be provided with the use of *dynamical graphs* [HS12]. Examples of these instances include social groups in which interactions are determined by *human proximity* [Cat+10; EP06] or *face-to-face interactions* [Bar+13; BC13], biological cells in which edges model gene or protein interactions [PSS10; Leb+10] and transportation networks [PS11; Bor+13; Ban+15], representing how different locations are connected among themselves.

The inclusion of an additional degree of freedom representing time makes it particularly challenging to agree on the most basic concepts concerning temporal graphs and no general consensus exists on several definitions and nomenclature [Hol15]. In the remainder we consider the following definition of temporal graph, sometimes referred to as *snapshot graph* [RC18].

A snapshot temporal graph (picture taken from [Muc+10])



Definition 9.1 (Snapshot graph)**.** *A* snapshot graph *is a temporal graph representation, composed by an ordered sequence of* $T \in \mathbb{N}$ *graph snapshots* $\{\mathcal{G}_1(\mathcal{V}_1, \mathcal{E}_1), \ldots, \mathcal{G}_T(\mathcal{V}_T, \mathcal{E}_T)\}$ *in which* $(ij) \in \mathcal{E}_t$ *indicates that nodes* i *and* j *are connected at time* t*. To a temporal graph is associated a sequence of adjacency matrices* $\{A^{(1)}, \ldots A^{(T)}\}$.

A *snapshot graph* is, by all means, a *multiplexed* graph (introduced in Chapter 1) in which each level corresponds to a snapshot. Note that in Definition 9.1 time is discrete and for each t, $\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)$ is a static graph.

With the basic notions concerning dynamical graphs being laid out, we take into account how the problem of CD can be formulated in its dynamical version, allowing us to extend our results from Part II to this setting.

9.1 COMMUNITY DETECTION IN DYNAMICAL GRAPHS

The problem of *dynamical community detection* (DCD) can be informally defined as the task of performing CD on a dynamical graph, assigning a class label to each node as a function of time. The underlying assumption motivating DCD is that, as the graph evolves, also the communities change their structure. As a practical example one can think of a *citation network* in which,

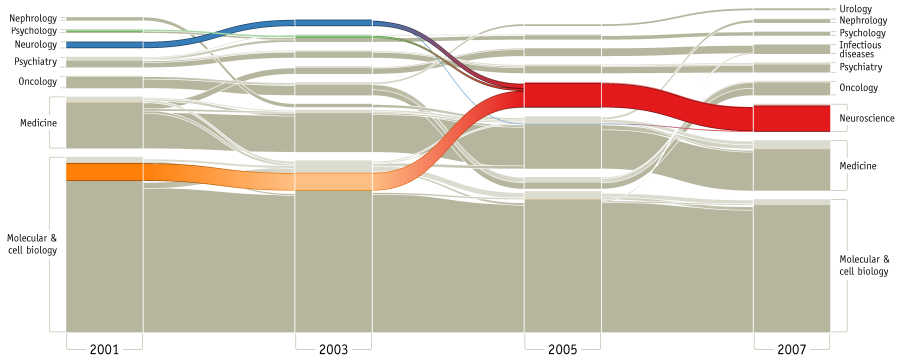


Figure 9.1: Alluvial diagram mapping change in scientific research in the years from 2001 to 2007. Each block corresponds to a field and its height is proportional to the number of citations occurred in that field.
Source: doi.org/10.1371/journal.pone.0008694.g003.

for instance, an edge[1] $(ij) \in \mathcal{E}_t$ means that author i cited author j in year t. The community structure in this graph is determined by the authors' research interests which may change over time, thus altering the community structure. A practical representation of this process is shown in Figure 9.1 (taken from [RB10]) in which changes in the community structure are evidenced with the help of an *alluvial diagram*.

As a consequence of the communities being expected to evolve smoothly across time [RC18; CR19], high correlations are observed in the community labels at different times. A major advantage of DCD with respect to CD is the possibility to exploit such correlations, hence using information that may come from past (or future) realizations of the graph to determine the class structure at a given time.

In order to formally address DCD, it is necessary to clearly frame the problem from a mathematical perspective. In Part II we insisted on how providing a definition to CD is a generally hard task and, in the dynamical case, the problem of defining communities is even more exacerbated by the complexity that rules the dynamical evolution of the graph. As a consequence, even the most expert authors in the field [HS12; RC18; CR19] have struggled to provide a unified view on the problem of DCD that, in general, may have very different definitions, strategies as well as objectives. For this reason, we do not attempt in this chapter to provide a (even incomplete) review of DCD: given the diversity of the different approaches, in many cases, results cannot be compared on even grounds. On the opposite, in the remainder we will specifically consider the problem of label inference from the *dynamical degree corrected stochastic block model* (DDCSBM) [Xu15] which is a generalization of the *degree corrected stochastic block model* (DCSBM) studied in Part II. Like in the static case, this model has the important advantage to provide a great mathematical control, motivating the reason of our choice.

1 For the sake of precision, we should denote $(ij) \in \mathcal{E}_{\mathrm{d},t}$ since the natural representation of this system is obtained with a directed network.

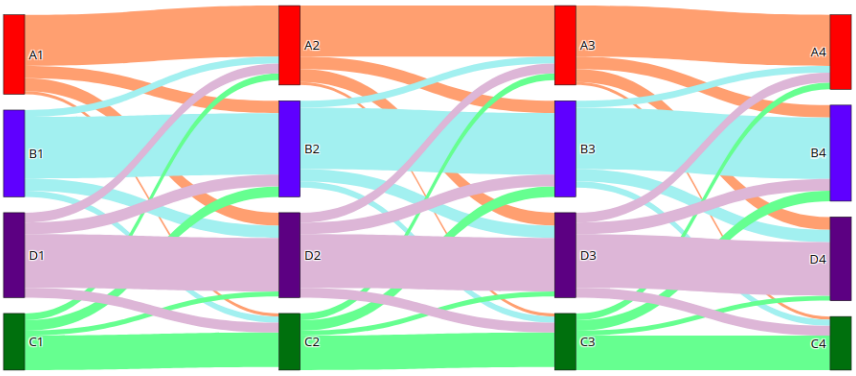


Figure 9.2: Alluvial diagram representing the dynamics of label vector obtained according to Definition 9.2. In this example $k = 4$, $T = 4$. The rows (and the colour code) corresponds to the four classes, while the columns to the time snapshots.

9.1.1 THE DYNAMICAL DEGREE CORRECTED STOCHASTIC BLOCK MODEL

In this section we introduce the DDCSBM together with its conjectured detectability threshold. First of all, let us settle some notation. We denote with $T \in \mathbb{N}$ the number of snapshots and we let $\{\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)\}_{t=1,\dots,T}$ be a sequence of unweighted and undirected graphs, each with n nodes. At time step t, $\mathcal{E}_t$ and $\mathcal{V}_t$ denote the set of edges and nodes, respectively, which form $\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)$, with $\mathcal{V}_t \cap \mathcal{V}_{t'} = \emptyset$, for $t' \neq t$: each node has T copies, each copy being a different object. We denote with i_t, for $1 \leq i \leq n$ and $1 \leq t \leq T$, a node in $\mathcal{V}_t$. We call $A^{(t)} \in \{0,1\}^{n\times n}$ the symmetric adjacency matrix of $\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)$ and $D^{(t)}$ its associated degree matrix. We now detail the generative model for $\{\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)\}_{t=1,\dots,T}$.

First of all the label dynamics must be specified. Let $\ell_{i_t} \in \{1,\dots,k\}$ be the label of node i_t, *i.e.* the label of node i at time t and let $\boldsymbol{\pi} \in \mathbb{R}^k$ be the (time independent) vector encoding the expected fraction of nodes belonging to each class. The labels are updated across time as per Definition 9.2.

Definition 9.2 (Label dynamics). *Given three positive integers k, n, T and a vector $\boldsymbol{\pi} \in \mathbb{R}^k$ satisfying $\mathbf{1}_k^T \boldsymbol{\pi} = 1$, the vector $\boldsymbol{\ell} \in \{1,\dots,k\}^{nT}$ is defined as follows. First, the entries $\{\ell_{i_{t=1}}\}_{i=1,\dots,n}$ are initialized by assigning random values with probability proportional to the class sizes: $\mathbb{P}(\ell_{i_1} = a) = \pi_a$. The remaining entries labels are then obtained for $2 \leq t \leq T$ according to*

The label dynamics is a Markov process

$$\ell_{i_t} = \begin{cases} \ell_{i_{t-1}} & \text{w.p. } \eta \\ a & \text{w.p. } (1-\eta)\pi_a, \;\; a \in \{1,\dots,k\}, \end{cases} \tag{9.1}$$

i.e., *the label of node i_t is maintained with probability η and otherwise reassigned at random with probability $1-\eta$.*

Note that, according to Definition 9.2, a proportion of the reassigned nodes from time t will be affected the same labels at time $t+1$. It is easy to verify from Definition 9.2 that the expected sizes of the clusters are constant across time, in fact:

$$\begin{aligned}\frac{\mathbb{E}\left[|\mathcal{V}_{t,a}|\right]}{n} &= \frac{1}{n}\mathbb{E}\left[\eta|\mathcal{V}_{t-1,a}| + \sum_{b=1}^{k}|\mathcal{V}_{t-1,b}|(1-\eta)\pi_a\right] \\ &= \eta\frac{\mathbb{E}\left[|\mathcal{V}_{t-1,a}|\right]}{n} + (1-\eta)\pi_a. \end{aligned} \tag{9.2}$$

where $\mathcal{V}_{t,a}$ is the set of nodes in $\mathcal{V}_t$ with label a. At time $t=1$, by hypothesis, $\mathbb{E}[|\mathcal{V}_{1,a}|] = n\pi_a$. Plugging this relation in Equation (9.2), one obtains with a recursive argument that for all t, $\mathbb{E}[|\mathcal{V}_{t,a}|] = n\pi_a$. Figure 9.2 shows the alluvial graphs of a typical realization of $\boldsymbol{\ell}$ for $k, T = 4$. Given the generative model of the label vector, we now provide the definition of the DDCSBM.

The dynamical degree corrected stochastic block model

Definition 9.3 (Dynamical degree corrected stochastic block model)**.** *Let $T \in \mathbb{N}$ be the number of instances of snapshots graph with n nodes. Denote with $\boldsymbol{\ell} \in \{1,\dots,k\}^{nT}$ the class label vector, generated according to Definition 9.2, where k is the number of classes. Further let $C \in \mathbb{R}^{k\times k}$ be a symmetric matrix with positive elements. and $\theta \in \Theta = [\theta_{\min}, \theta_{\max}]$ be a random variable that encodes the intrinsic node connectivity, distributed according to ν, satisfying $\int_\Theta d\nu(\theta) = 1$ (normalization), $\mathbb{E}[\theta] = \int_\Theta \theta\, d\nu(\theta) = 1$, $\mathbb{E}[\theta^2] = \int_\Theta \theta^2\, d\nu(\theta) \equiv \Phi = O_n(1)$. For each node, θ_i is drawn independently at random from the distribution ν.*

The entries of the (symmetric) adjacency matrix $A^{(t)}$ of $\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)$ are set to one independently and independently across time, with probability:

$$\mathbb{P}(A_{ij}^{(t)} = 1) = \min\left(\theta_i\theta_j\frac{C_{\ell_{i_t},\ell_{j_t}}}{n}, 1\right)$$

and are equal to zero otherwise.

In words, given the label vector $\boldsymbol{\ell}$, the DDCSBM generates the elements of $\{\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)\}_{t=1,\dots,T}$ independently at random from the static DCSBM with the class labels that change across time. This model makes strong assumptions on the dynamics, keeping fixed the matrix C, the number and size of the communities, as well as the parameter η, but it is however capable to generate a snapshot graph with labels correlated across time.

Before proceeding any further, let us make a remark of the independence assumption of each graph across time.

Remark 9.1 (Graph independence across time)**.** *In the DDCSBM the instances $\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)$ are generated* independently *across time. Let us translate this assumption in simpler terms. In a practical setting, we can define a time scale, say τ, which determines the speed at which the graph evolves. Assuming that the snapshots of the graph are independent means to assume that $\tau \ll 1$,* i.e. *that the typical time needed for the graph edges to rearrange is much smaller than the distance between two successive snapshots. This assumption is hence adequate when the snapshots are sufficiently far in time.*

Related to Remark 9.1, it has to be noted that a second time-scale can be defined, describing the speed at which the community structure evolves. This time scale depends on η and goes to infinity for $\eta = 1$ (the community labels are *frozen*) and to zero for $\eta \to 0$ (the communities are completely reconfigured between one snapshot and the successive one). The fundamental hypothesis introduced by the DDCSBM consists in assuming that the graph evolves at a much faster rate than the communities, so that, while $\tau \to 0$ (due to the independence assumption), $\eta = O(1)$.

Our objective is to infer the vector $\boldsymbol{\ell} \in \{1, \dots, k\}^{nT}$, hence to assign to each node at each time the appropriate community label. Like in the static case, also for DDCSBM there exists a detectability threshold below which inference is not asymptotically feasible. In particular, in [Gha+16] the authors retraced the steps made by [Dec+11] in the static regime, deriving the (asymptotically exact) *belief propagation* (BP) equations to perform inference in the *dynamical stochastic block model* (DSBM) (in which $\boldsymbol{\theta} = \mathbf{1}_n$) to then determine the instability condition of BP. In the setting of k classes of equal size with $C = c_{\text{out}}\mathbf{1}_k\mathbf{1}_k^T + (c_{\text{in}} - c_{\text{out}})I_k$, $c = \frac{1}{k}C\mathbf{1}_k$ and defining α as in the static case, *i.e.* $\alpha = (c - c_{\text{out}})/\sqrt{c}$, the authors of [Gha+16] conjectured that, for $T \to \infty$, DCD is feasible if $\alpha > \alpha_c(\infty, \eta)$, where $\alpha_c(\infty, \eta)$ is the unique value of $\bar{\alpha} > 0$ for which the largest eigenvalue of

$$M_\infty(\bar{\alpha}, \eta) = \begin{pmatrix} \bar{\alpha}^2 & 2\eta^2 \\ \bar{\alpha}^2 & \eta^2 \end{pmatrix} \tag{9.3}$$

is equal to one. In [Gha+16] the authors further gave clues to generalize their result for T finite but did not provide an explicit expression to $\alpha_c(T, \eta)$. In Section 9.2 we extend the conjecture of [Gha+16], providing the explicit expression to $\alpha_c(T, \eta)$ and generalizing it to the DDCSBM setting.

With this fundamental property of the DDCSBM being settled, let us now present the main contributions and challenges considered in this chapter.

9.1.2 RELATED WORKS AND CONTRIBUTIONS

Our main contribution consists in the design of a new, efficient algorithm to perform SC in dynamical graphs generated from the DDCSBM. Albeit SC in dynamical graphs is much less studied than it is on static graphs, recently it has gained more and more interest from the scientific community and SC algorithms have also been designed to operate in the dynamic regime. The main goal of DCD is to exploit the correlation of the labels across time in order to improve the clustering performance.

One of the most common approaches is given by *evolutionary spectral clustering* [Chi+07; Qin+16; Chi+09; Liu+18; XKHI14] in which the community assignment is defined for all t, optimizing $\mathscr{Q}_t^{\text{ev}}$, defined for all $t \geq 2$ as:

$$\mathscr{Q}_t^{\text{ev}}(\boldsymbol{\ell}_{(t)}, \boldsymbol{\ell}_{(t-1)}) = h\mathscr{Q}_t^{\text{snap}}(\boldsymbol{\ell}_{(t)}) + (1-h)\mathscr{Q}_{t,t-1}^{\text{smooth}}(\boldsymbol{\ell}_{(t)}, \boldsymbol{\ell}_{(t-1)}), \tag{9.4}$$

where h is a parameter between 0 and 1, $\mathscr{Q}_t^{\text{snap}}$ and $\mathscr{Q}_{t,t-1}^{\text{smooth}}$ are two quality functions evaluating the goodness of a partition on a single snapshot and imposing a smooth change between $\boldsymbol{\ell}_{(t)} = \{\ell_{i_t}\}_{i=1,\dots,n}$ and $\boldsymbol{\ell}_{(t-1)}$, respectively. More practically, $\mathscr{Q}_t^{\text{snap}}$ can be, for instance, the *modularity*, the RCut, or some other cost function introduced in Section 4.1; an example of $\mathscr{Q}_{t,t-1}^{\text{smooth}}$, instead, is the overlap which corresponds to large values if $\boldsymbol{\ell}_{(t)}$ and $\boldsymbol{\ell}_{(t-1)}$ are similar, while it tends to zero if the community labels are completely rearranged. Tuning the value of h, one can enforce higher (or smaller) levels of smoothness in the label evolution.

Spectral clustering allows one to obtain a relaxed solution of the optimization of Equation (9.4). Beyond all the weaknesses of the optimization approach that we extensively discussed in Part II, it has to be noted that: (i) the function $\mathscr{Q}_t^{\text{ev}}$ does not contain any information from all $t' > t$ (it only depends on t and $t-1$); (ii) the resulting relaxation does not, in general, allow one to achieve the detectability threshold and are not adapted for sparse graphs; (iii) smoothness is often included in a very heuristic way, consequently leading to suboptimal algorithms.

A different, but somewhat related approach consists in performing SC using as an input a modified adjacency matrix obtained by averaging the last realizations of the adjacency matrix, with possibly a forgetting factor giving less weight to older times [PZ+19]. For instance, we can write

$$\tilde{A}^{(t)} = (1-h)\tilde{A}^{(t-1)} + hA^{(t)}.$$

Also in this case, clustering at time t does not exploit information from instances appearing after t. This method has however been studied in the sparse regime $c = O_n(1)$ by [KV20] in which the authors showed that for $\eta \to 0$ (the classes are supposed to be essentially frozen) and T sufficiently large, one obtains good clustering performances in the sparse regime, simply running Algorithm 4.1 on the matrix $M = \frac{1}{T}\sum_{t=1}^{T} A^{(t)}$. While this result is powerful, because it formally addresses the sparse regime, it essentially treats DCD as a perturbation to static CD and does not cover the more interesting regime in which communities can evolve rapidly with time.

Averaging the adjacency matrix simply corresponds to the choice $h = 1$ in $\tilde{A}^{(t)} = (1-h)\tilde{A}^{(t-1)} + hA^{(t)}$

To the best of our knowledge, the work of [Gha+16] provides the only existing spectral algorithm properly treating both sparsity and small label persistence. In the spirit of [Krz+13], the proposed method arises from a linearization of the BP, which is capable of obtaining non-trivial partitions as soon as theoretically possible. However, their resulting dynamical non-backtracking matrix depends on an *a priori* unknown parameter,[2] so the algorithm is practically inapplicable.

As an answer to these limitations, we propose a new spectral algorithm adapted to the sparse regime, which is able to detect communities even under little (or no) persistence in the community labels and which benefits from

2 In order to design their dynamical non-backtracking matrix, the average number of connections among nodes in the same and across communities must be known.

persistence to improve classification performance over a static algorithm run independently at each time-step. Specifically,

- We introduce a dynamical Bethe-Hessian matrix which, for $k = 2$ classes of equal size, retrieves non-trivial communities as soon as theoretically possible for any T, η. As a by-product, we offer new results on the spectrum of the dynamical non-backtracking of [Gha+16].
- We provide an algorithm applicable to any graph with $k \geq 2$ communities of arbitrary sizes.[3] The resulting Algorithm 9.1 is part of the `CoDeBetHe.jl` package.

9.2 DETECTABILITY THRESHOLD FOR FINITE T

In this section we formulate our generalization for finite T of the claim of [Gha+16] concerning the *detectability threshold* in the DSBM, further considering its extension to the DDCSBM setting. For convenience, we assume in the remainder the case of $k = 2$ classes of equal sizes, leaving to Section 9.4 the discussion of how our results can be straightforwardly generalized to the $k > 2$ setting.

In [Gha+16] the authors introduce a threshold $\alpha_c(T, \eta)$, however not explicitly define,[4] below which ($\alpha < \alpha_c(T, \eta)$) community detection is not feasible. We go here beyond [Gha+16] by providing an explicit value for $\alpha_c(T, \eta)$ for all finite T as in the following claim, whose details are treated in Appendix C.1

Claim 9.1 (Detectability threshold in the DDCSBM). *Let $\{\mathcal{G}_t(\mathcal{V}_t, \mathcal{E}_t)\}_{t=1,\dots,T}$ be a sequence of graphs generated from a DDCSBM with $k = 2$ classes of equal size as per Definition 9.3 with $C_{ab} = c_{\text{in}}$ if $a = b$ and c_{out} otherwise. Let $\alpha = (c - c_{\text{out}})\sqrt{\Phi/c}$, then the labels of the DDCSBM can be inferred with an efficient algorithm only if $\alpha > \alpha_c(T, \eta)$, where $\alpha_c(T, \eta)$ is the only positive $\bar{\alpha}$ for which the largest eigenvalue of $M_T(\bar{\alpha}, \eta)$ is equal to one, where*

$$M_T(\bar{\alpha}, \eta) = \begin{pmatrix} M_d & M_+ & 0 & \dots & 0 \\ M_- & M_d & \ddots & \dots & 0 \\ 0 & M_- & \ddots & M_+ & 0 \\ \vdots & \vdots & \ddots & M_d & M_+ \\ 0 & 0 & \dots & M_- & M_d \end{pmatrix}, \text{ with } \begin{cases} M_d = \begin{pmatrix} 0 & 0 & 0 \\ \eta^2 & \bar{\alpha}^2 & \eta^2 \\ 0 & 0 & 0 \end{pmatrix}, \\ M_+ = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & \bar{\alpha}^2 & \eta^2 \end{pmatrix}, \\ M_- = \begin{pmatrix} \eta^2 & \bar{\alpha}^2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}. \end{cases}$$

3 The algorithm *a priori* requires that η be known; otherwise, η can be estimated through cross-validation.

4 Precisely, quoting the authors, this is as far as $\alpha_c(T, \eta)$ is defined: "We can compute the corresponding finite-time threshold for a fixed T by diagonalizing a $(3T - 2)$-dimensional matrix, where we have a branching process with states corresponding to moving along spatial, forward-temporal, or backward-temporal edges at each time step".

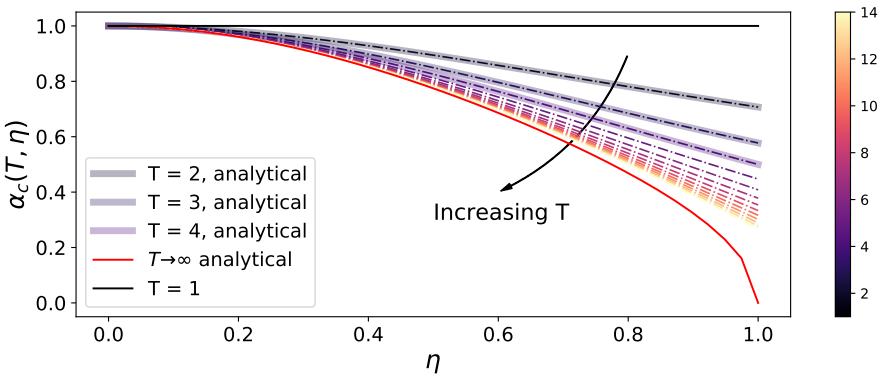


Figure 9.3: Position of $\alpha_c(T,\eta)$ as a function of η (x axis) and T (color code). The black and red solid lines correspond to the analytical values of $\alpha_c(T,\eta)$, for $T = 1$ and $T = \infty$, respectively. The dashed dotted lines are the position of $\alpha_c(T,\eta)$ computed numerically, and the thick solid pale lines are the analytical values of $\alpha_c(T,\eta)$ for $T \in \{2,3,4\}$.

The expression of $\alpha_c(T,\eta)$ can be computed analytically for $T = 2,3,4$ and $T \to \infty$:

$$\alpha_c(T=2,\eta) = \left(1+\eta^2\right)^{-\frac{1}{2}}$$

$$\alpha_c(T=3,\eta) = \sqrt{2}\left(2+\eta^4+\eta^2\sqrt{8+\eta^4}\right)^{-\frac{1}{2}}$$

$$\alpha_c(T=4,\eta) = \sqrt{2}\left(2+\eta^2+\eta^6+\eta\sqrt{\eta^8+2\eta^4+8\eta^2+5}\right)^{-\frac{1}{2}}$$

$$\alpha_c(\infty,\eta) = \left(\frac{1+\eta^2}{1-\eta^2}\right)^{-\frac{1}{2}}.$$

For other values of T, $\alpha_c(T,\eta)$ is best evaluated numerically.

High label correlation and many graph observations allow one to solve harder CD problems

The value of $\alpha_c(T,\eta)$ is shown as a function of η for different T in Figure 9.3. Let us make some basic observations on the values of $\alpha_c(T,\eta)$. For all T: (i) if $\eta = 0$ (no correlation among the labels), one recovers $\alpha_c = 1$, the transition's position in the static DCSBM [GLM15], as expected; (ii) if $\eta = 1$, $\alpha_c = 1/\sqrt{T}$, the static threshold obtained by averaging the adjacency matrix over its T independent and identically distributed realizations. This result should be related to [KV20] that indeed suggests to used the averaged adjacency matrix to perform SC when $\eta \to 1$. We also numerically confirm that for all T, $\alpha_c(T,\eta)$ is a decreasing function of η: higher label persistence allows to solve harder problems.

9.3 MAIN RESULT

This section develops a "dynamical" Bethe-Hessian matrix associated to the graph sequence $\{\mathcal{G}_t(\mathcal{V}_t,\mathcal{E}_t)\}_{t=1,\dots,T}$, for which we show there exists *at least*

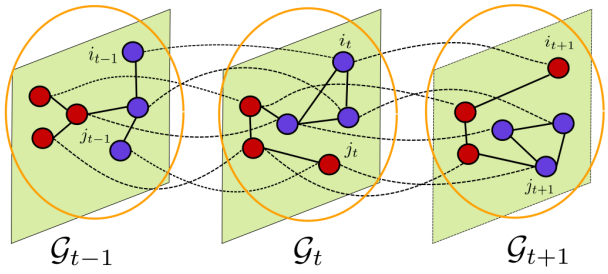


Figure 9.4: Three successive snapshots of the a aggregate graph $\mathcal{G}(\mathcal{V},\mathcal{E})$. Classes are emphasized by node colors and can evolve with time. Network edges, that change over time, are indicated in solid lines, while "temporal edges" in dashed lines connect each graph to its temporal neighbors. Nodes of a common time step are circled in orange.

one eigenvector (recall that $k = 2$ classes so far) strongly aligned to the community labels if $\alpha > \alpha_c(T,\eta)$, thereby allowing us to design a CD algorithm that achieves the detectability threshold. The eigenvectors containing information can be as many as T, but only one of them is guaranteed to exist when $\alpha > \alpha_c(T,\eta)$ and it can alone reconstruct communities.

Before introducing our main result, it is convenient to introduce the *aggregate graph* $\mathcal{G}(\mathcal{V},\mathcal{E})$, defined as follows.

Definition 9.4 (Aggregate graph). *Letting $\{\mathcal{G}_t\}_{t=1\dots T}$ be a sequence of graphs generated as per Definition 9.3, $\mathcal{G} = \mathcal{G}(\mathcal{V},\mathcal{E})$ is the graph with $\mathcal{V} = \cup_{t=1}^{T}\mathcal{V}_t$ and $\mathcal{E} = \left(\cup_{t=1}^{T}\mathcal{E}_t\right) \cup \left(\cup_{t=1}^{T-1}\cup_{i=1}^{n}(i_t, i_{t+1})\right)$. The adjacency and degree matrices of $\mathcal{G}(\mathcal{V},\mathcal{E})$ are denoted with $A, D \in \mathbb{N}^{nT\times nT}$, respectively. In other words, the graphs $\mathcal{G}_t(\mathcal{V}_t,\mathcal{E}_t)$ are joined adding extra edges between the nodes i_t and their temporal neighbours $i_{t\pm1}$.*

Figure 9.4 gives a visual representation of the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$.

9.3.1 THE DYNAMICAL BETHE-HESSIAN MATRIX

As in Chapter 5, our approach exploits a statistical physics analogy between the modelling of spontaneous magnetization of spins with *ferromagnetic* interaction and the modelling of communities of nodes in sparse graphs. We define the following *Hamiltonian*[5] on the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$.

$$\mathcal{H}_{\xi,h}(\boldsymbol{s}) = -\left(\sum_{t=1}^{T}\sum_{(i_t,j_t)\in\mathcal{E}_t}\mathrm{ath}(\xi)\, s_{i_t}s_{j_t} + \sum_{t=1}^{T-1}\sum_{i_t\in\mathcal{V}_t}\mathrm{ath}(h)\, s_{i_t}s_{i_{t+1}}\right) \tag{9.5}$$

5 Note that a very similar cost function has been used in [Muc+10] to find communities in dynamical graphs as the solution of an optimization problem.

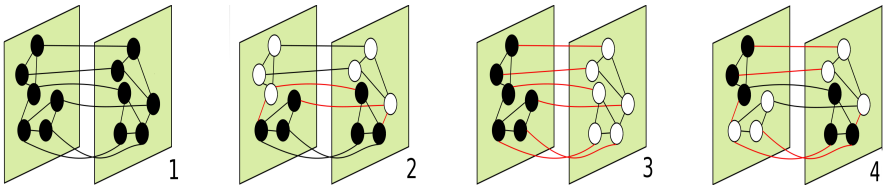


Figure 9.5: Sketch of the 4 stable modes for two communities and $T = 2$. In black we indicate the direction $s_i > 0$, in white $s_i < 0$. The red edges correspond to the frustrated edges connecting spins with opposite direction.

Here, the coupling constants $\xi, h \in [0, 1)$ modulate the interaction among nodes at time t and between the same node at time instants t and $t+1$, respectively and appear inside inverse hyperbolic tangents for notational ease. Intuitively, the spin vector $\boldsymbol{s}$ can be mapped to the class affiliation vector $\boldsymbol{\sigma} \in \{\pm 1\}^{nT}$, in which $\sigma_{i_t} = 1$ if $\ell_{i_t} = 1$ and $\sigma_{i_t} = -1$ otherwise. The first term in the main parenthesis of (9.5) favors configurations in which neighboring nodes have the same label, while the second term favors configurations in which the label is kept across successive time instants, enforcing persistence in the community evolution.

The mesoscale structure of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ determines the appearance of some local minima in $H_{\xi,h}(\boldsymbol{s})$ like, for instance, the configuration $\boldsymbol{s} = \boldsymbol{\sigma}$. A sketch of such minima of $H_{\xi,h}(\boldsymbol{s})$ is provided for $T = 2$ in Figure 9.5. The lowest energy state corresponds to $\boldsymbol{s} = \mathbf{1}_{nT}$: this is the non-informative *ferromagnetic configuration*. Similarly, mode 3 of Figure 9.5 groups together nodes in the same community and is equally useless for reconstruction. On the opposite, modes 2 and 4 of Figure 9.5 divide the nodes according to the class structure of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ and can be used for community reconstruction. In general, for k classes and $T > 2$ time frames, kT such local minima arise, mixing together time and class clusters. Note importantly that mode 1 always has a lower energy than mode 3 and mode 2 a lower energy than mode 4. However, the ordering of energies of modes 2 and 3 is in general not *a priori* known. We will further comment on this remark which has important consequences for the subsequent analysis as well as for the design of our proposed community detection algorithm.

The eigenvectors corresponding to the smallest eigenvalues of the Bethe-Hessian matrix (Definition 2.2) associated to the Hamiltonian of Equation (9.5) are strongly correlated with the configurations appearing in Figure 9.5 and can therefore be used to perform DCD. From the results described in Section 2.3.2, the dynamical Bethe-Hessian matrix $H_{\xi,h} \in \mathbb{R}^{nT \times nT}$ associated to the Hamiltonian of Equation (9.5) is defined as follows

$$\left(H_{\xi,h}\right)_{i_t, j_{t'}} = \begin{cases} \left(\frac{\xi^2 D^{(t)} - \xi A^{(t)}}{1-\xi^2} + \frac{1+h^2(\phi_t - 1)}{1-h^2} I_n\right)_{ij} & \text{if } t = t' \\ \left(-\frac{h}{1-h^2} I_n\right)_{ij} & \text{if } t = t' \pm 1, \end{cases} \tag{9.6}$$

The dynamical Bethe-Hessian matrix

where $\phi_t = 1$ if $t = 1$ or $t = T$ and $\phi_t = 2$ otherwise. The aforementioned lack of a precise knowledge of the relative position of the informative modes

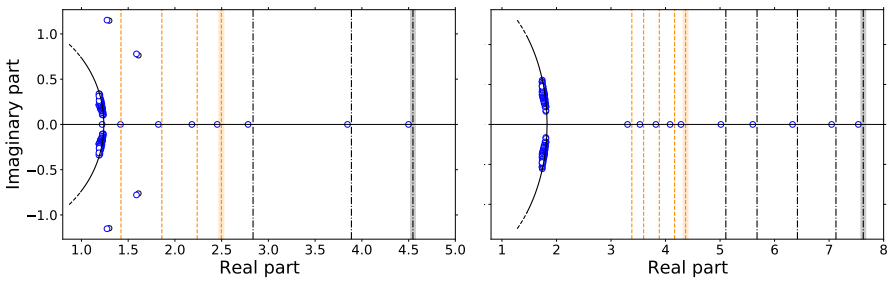


Figure 9.6: The 150 eigenvalues of $B_{\xi,h}$ with largest real part, for $n = 10\,000$, $T = 5$, $\eta = 0.4$, $c = 10$, $c_{\text{out}} = 4$, $\theta_i \sim [\mathcal{U}(3,10)]^3$. **Left**: $\xi = 0.2$, $h = 0.9$. **Right**: $\xi = 0.4$, $h = 0.7$. The orange and black dashed lines are the theoretical positions of the eigenvalues forming the *informative* and *uninformative* families, respectively. The thickest of these lines correspond to the leading eigenvalue of the respective family. The imaginary eigenvalues are represented with circles in the complex plane. The solid black line is a part of the circle of radius $L_{\xi,h}$.

in the energy spectrum of the Hamiltonian hampers the identification of the position of the corresponding informative eigenvectors of $H_{\xi,h}$, which is of major importance to design a SC algorithm based on $H_{\xi,h}$.

9.3.2 COMMUNITY DETECTABILITY WITH THE DYNAMIC BETHE-HESSIAN

We thus now turn to our main result centred on the question of appropriately choosing a pair (ξ, h) which ensures non-trivial community detection with $H_{\xi,h}$ as soon as $\alpha > \alpha_c(T,\eta)$.

In order to determine the spectral properties of $H_{\xi,h}$, we study the eigenvalues of the weighted non-backtracking matrix $B_{\xi,h}$, to then exploit the relation between the two spectra as per Theorem 2.1 and Claim 3.2. Recalling the definition of the weighted non-backtracking matrix used in Section 3.2.3, the expression of $B_{\xi,h}$ on the aggregate graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ reads

The dynamical non-backtracking matrix

$$\forall (ij),(kl) \in \mathcal{E}_{\text{d}}, \quad \left(B_{\xi,h}\right)_{(ij)(kl)} = \delta_{jk}(1-\delta_{il})\,\omega_{kl}, \tag{9.7}$$

where $\mathcal{E}_{\text{d}}$ is the set of directed edges of the aggregate graph $\mathcal{G}(\mathcal{V},\mathcal{E}), \omega_{kl} = \xi$ is (kl) is a spatial edge and $\omega_{kl} = h$ if it is a temporal edge. The choice of a proper parametrization (ξ, h) to perform non-trivial DCD with the dynamical Bethe-Hessian matrix comes from the following central claim describing the spectrum of $B_{\xi,h}$, whose detailed derivation is provided in Appendix C.2.

Claim 9.2 (Spectrum of $B_{\xi,h}$ on sparse DDCSBM graphs). *Let $\mathcal{G}(\mathcal{V},\mathcal{E})$ be a graph generated as per Definition 9.4 with $k = 2$ communities, $C_{a=b} = c_{\text{in}}$, $C_{a\neq b} = c_{\text{out}}$ and $\gamma = (c_{\text{in}} - c_{\text{out}})/(c_{\text{in}} + c_{\text{out}})$. Recalling the definition of $M_T(\cdot,\cdot)$ from Claim 9.1, then, with high probability, in the $n \to \infty$ limit, the spectrum of $B_{\xi,h}$ can be described as follows:*

- *the complex eigenvalues forming the bulk of $B_{\xi,h}$ are bounded by a disk in the complex plane of radius $L_{\xi,h} = \sqrt{\lambda_1^{\downarrow|\cdot|}(M_T(\sqrt{c\Phi\xi^2}, h))}$;*
- *all the eigenvalues of $B_{\xi,h}$ of magnitude larger than $L_{\xi,h}$ are isolated and are asymptotically close to one of the eigenvalues of either $M_T(\sqrt{c\Phi\xi}, \sqrt{h})$ (in which case the eigenvectors are not correlated to the community labels) or $M_T(\sqrt{c\Phi\xi\gamma}, \sqrt{\eta h})$ (in which case the eigenvectors are correlated to the community labels).*

Figure 9.6 confirms numerically Claim 9.2. We choose to compute only the 150 eigenvalues with largest real part to keep a reasonable computational time, while having a large value of n. From Claim 9.2, the position of the isolated and bulk eigenvalues of $B_{\xi,h}$ is defined for all (ξ, h). Note that two *families* of isolated eigenvectors appear in the spectrum of $B_{\xi,h}$: one is the *informative* family (related to the configurations 2, 4 of Figure 9.5) and one is the *uninformative* family (related to configurations 1, 3 of Figure 9.5). The ordering of the eigenvalues belonging to these two families is a priori unknown and depends on ξ, h. Let us make a remark on an interesting behavior of the isolated eigenvalues of $B_{\xi,h}$.

Remark 9.2 (Complex isolated eigenvalues in the spectrum of $B_{\xi,h}$)**.** *Note that the isolated eigenvalues of $B_{\xi,h}$ are asymptotically close to the eigenvalues of the matrix $M_T(\cdot,\cdot)$. This matrix is real and non-negative, but it is not symmetric. Consequently, its leading eigenvalue is certainly real (due to Perron-Frobenius theorem), while the subsequent eigenvalues are potentially complex. Although we cannot offer a clear interpretation for the complex nature of some of these isolated eigenvalues, our study is experimentally verified to hold also in this case as shown in the left plot of Figure 9.6.*

Based on Claim 9.2, we now provide two choices of the parametrization (ξ, h) that allow one to devise a SC algorithm based on the Bethe-Hessian matrix to perform DCD as soon as $\alpha > \alpha_c(T, \eta)$.

The optimal Bethe-Hessian parametrization

The optimal parametrization is obtained from the relaxation of the asymptotically Bayes optimal BP equations

The first proposed parametrization that allows one to perform non-trivial DCD as soon as $\alpha > \alpha_c(T, \eta)$ consists in choosing $(\xi, h) = (\gamma, \eta)$. Recalling the results of Chapter 5, we note that, in this setting, $\gamma = \zeta_2^{-1} + o_n(1)$, the optimal parametrization of the static Bethe-Hessian matrix. The fact that γ is (asymptotically) the inverse of ζ_2 (and not ζ_2), is just due to a notational convenience: in fact in Chapter 5, $r \geq 1$, while here $\xi \leq 1$.

First of all we want to show that the matrix $B_{\gamma,\eta}$ can be used for community reconstruction. This is a straightforward consequence of Claim 9.2: letting $\xi = \gamma$ and $h = \eta$, we obtain that the leading informative eigenvalue is equal to $\lambda_{\text{info},1} = \lambda_1^{\downarrow|\cdot|}(M_T(\alpha, \eta))$, while the radius of the bulk is equal to $L_{\lambda,\eta} = \sqrt{\lambda_1^{\downarrow|\cdot|}(M_T(\alpha, \eta))} = \sqrt{\lambda_{\text{info},1}}$. By definition, if $\alpha > \alpha_c(T, \eta)$, then $\lambda_{\text{info},1} > 1$, therefore $\lambda_{\text{info},1} > L_{\lambda,\eta}$. So for all $\alpha > \alpha_c(T, \eta)$, $\lambda_{\text{info},1}$ is an

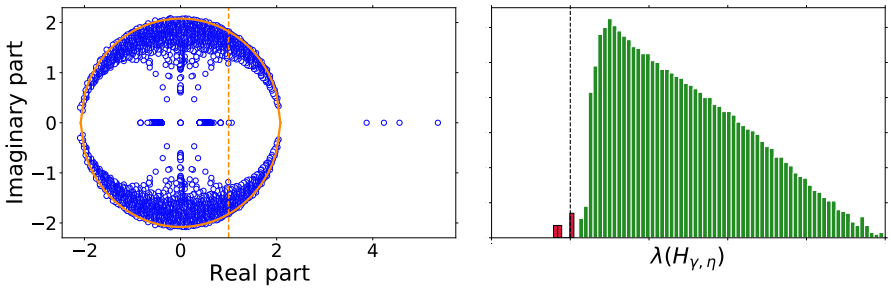


Figure 9.7: **Left**: spectrum in the complex plane of $B_{\gamma,\eta}$ The vertical line is at 1, while the circle has radius $L_{\gamma,\eta}$. **Right**: histogram of the eigenvalues of $H_{\gamma,\eta}$. The vertical line is at $\lambda = 0$. For both simulations, $T = 2$, $\eta = 0.4$, $c = 6$, $c_{\text{out}} = 1$, $\boldsymbol{\theta} = \mathbf{1}_n$, $n = 2\,000$.

isolated eigenvalue in the spectrum of $B_{\lambda,\eta}$ and it contains information on the community labels.

The claim of optimality of the choice (γ, η) comes from the fact that, like in the static case, this is the natural parametrization obtained linearising the Bayes optimal BP equations. Let H be defined as follows:

$$H = \begin{pmatrix} \frac{1+\eta}{2} & \frac{1-\eta}{2} \\ \frac{1+\eta}{2} & \frac{1-\eta}{2} \end{pmatrix}$$

Letting $a, b \in \{1, 2\}$, the fixed point BP equations take the form [Gha+16, Equations 5, 6, 8]

$$m_{j_t,i_t}(a) = \frac{e^{-f_t(a)}}{Z_{j_t,i_t}} \left(\sum_b H_{ab}\, m_{i_t,i_{t+1}}(b) \right) \left(\sum_b H_{ab}\, m_{i_t,i_{t-1}}(b) \right) \prod_{l_t \in \partial i_t \setminus j_t} \sum_b C_{ab} m_{i_t,l_t}(b)$$

$$m_{i_{t+1},i_t}(a) = \frac{e^{-f_t(a)}}{Z_{i_{t+1},i_t}} \left(\sum_b H_{ab}\, m_{i_t,i_{t-1}}(b) \right) \prod_{l_t \in \partial i_t} \sum_b C_{ab} m_{i_t,l_t}(b)$$

where

$$f_t(a) = \frac{1}{n} \sum_{j \in \mathcal{V}_t} \sum_b C_{ab} m_{i_t,j_t}(b).$$

Expanding the above messages around the *trivial* fixed point $m_{j_t,i_t}(1) = 1/2 + \epsilon_{i_t,j_t}$, $m_{i_t,i_{t\pm1}}(2) = 1/2 - \epsilon_{i_t,i_{t\pm1}}$, we obtain

$$\epsilon_{j_t,i_t} = \eta(\epsilon_{i_t,i_{t-1}} + \epsilon_{i_t,i_{t+1}}) + \gamma \sum_{\ell_t \in \partial i_t \setminus j_t} \epsilon_{i_t,\ell_t}$$

$$\epsilon_{i_{t+1},i_t} = \eta \epsilon_{i_t,i_{t-1}} + \gamma \sum_{\ell_t \in \partial i_t} \epsilon_{i_t,\ell_t}.$$

These equations can be rewritten in synthetic form introducing exploiting the non-backtracking matrix

$$B_{\gamma,\eta} \boldsymbol{\epsilon} = \boldsymbol{\epsilon} + \mathrm{h.o.t}$$

In agreement with our empirical observations, we predict that the informative eigenvalue $\lambda_{\text{info},1}$ of $B_{\lambda,\eta}$ has a twin companion eigenvalue asymptotically close to one. Recalling that if $\alpha > \alpha_c(T,\eta)$, then the radius of the bulk of $B_{\gamma,\eta}$ is $L_{\gamma,\eta} > 1$, we conclude that this is an eigenvalue *inside* the bulk of $B_{\gamma,\eta}$ and isolated down to the detectability threshold, like in the static case. The left plot of Figure 9.7 confirms this observation, evidencing that, indeed, the matrix $B_{\gamma,\eta}$ has such unitary eigenvalue inside its bulk.

Applying Theorem 2.1 it is easily shown that the dynamical Bethe-Hessian matrix $H_{\gamma,\eta}$ has a zero eigenvalue as shown in the right plot of Figure 9.7. We summarize and formalize these results in the following claim.

Claim 9.3 (Spectral clustering with $H_{\gamma,\eta}$)**.** *Let $\mathcal{G}(\mathcal{V},\mathcal{E})$ be an aggregate graph obtained as in Definition 9.4. As $n \to \infty$, with high probability, the complex bulk eigenvalues of $B_{\gamma,\eta}$ are asymptotically bounded by a circle in the complex plane of radius $L_{\gamma,\eta} = \sqrt{\lambda_1^{\downarrow|\cdot|}(M_T(\alpha,\eta))}$.*

Besides, if $\alpha > \alpha_c(T,\eta)$, then $1 < L_{\lambda,\eta}$, 1 is an isolated eigenvalue of $B_{\lambda,\eta}$ and 0 is an isolated eigenvalue of $H_{\lambda,\eta}$, and the corresponding eigenvectors for both matrices are correlated to the vector of community labels.

The matrix $H_{\lambda,\eta}$ hence represents a natural extension of the optimal Bethe-Hessian discussed in Chapter 5 for static CD. While from a theoretical perspective Claim 9.3 may seem a sufficient result, from a practical viewpoint a fundamental piece of information is lacking: the location of the zero eigenvalue in the ordered set of eigenvalues of $H_{\gamma,\eta}$. We explained extensively in Part II the importance of actually being able to locate the informative eigenvalues of a matrix to design a SC algorithm and one of the big results of Claim 5.1 concerns indeed the position of the informative eigenvalues of the Bethe-Hessian. Unlike Claim 5.1, however, Claim 9.3 does not specify the location of the informative eigenvalues. This is because, as we anticipated already, the eigenvalues of *informative* and *uninformative* families described in Claim 9.2 can potentially be in a shuffled order, making it unpractical (if not impossible) to locate the informative eigenvalues without a prior knowledge of the values (γ,η). One may object that (γ,η) must be known to build $H_{\gamma,\eta}$. It has to be recalled, however, that (γ,η) are generally not known in advance and have to be estimated. The lack of knowledge of the position and value of the informative eigenvalues, however, makes it impossible to design an algorithm like 5.1 to estimate (γ,η) and perform DCD.

Before proceeding with the second possible parametrization that overcomes this problem, let us formulate a further remark on the matrix $B_{\gamma,\eta}$, relating our results to those of [Gha+16].

Remark 9.3 (Comparison with the spectral method of [Gha+16])**.** *In [Gha+16], the authors propose a SC method based on the matrix $B_{\gamma,\eta}$, obtained linearising the BP equations around their trivial fixed point as shown above. As we mentioned already in Section 9.1, this matrix needs to know the parameter γ in order to be defined and the related SC algorithm cannot be straightforwardly implemented since, as we just commented, γ is not easy to estimate.*

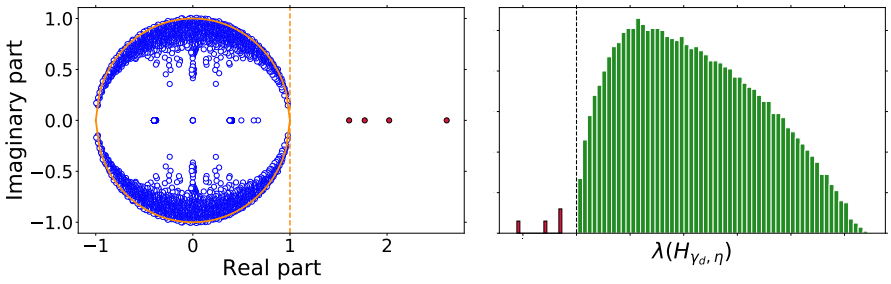


Figure 9.8: **Left**: spectrum in the complex plane of $B_{\gamma_d,\eta}$ The vertical line is at 1 and the circle has radius $L_{\gamma_d,\eta} = 1$ In red, the 4 isolated eigenvalues of $B_{\gamma_d,\eta}$ are evidenced. **Right**: histogram of the eigenvalues of $H_{\gamma_d,\eta}$. The vertical line is at $\lambda = 0$. For both simulations, $T = 2$, $\eta = 0.4$, $c = 6$, $c_{\text{out}} = 1$, $\boldsymbol{\theta} = \mathbf{1}_n$, $n = 2\,000$.

On top of this, in [Gha+16] the authors prescribed, for a general setting with k communities, to perform SC on the k dominant eigenvectors of $B_{\gamma,\eta}$. Claim 9.2 however clearly evidences (in the case $k = 2$) that the k largest eigenvalues are not necessarily informative. Consequently, using the k dominant eigenvectors to perform SC may lead to poor performance, as evidenced in Figure 9.10.

Let us now show that an alternative parametrization (ξ, h) equally allows one to perform DCD as soon as theoretically possible.

An alternative parametrization

Recall that in [SKZ14], SC is performed using H_r for $r = \sqrt{c\Phi}$

The second parametrization (ξ, h) to achieve detectability down to the threshold is inspired from the Bethe-Hessian of [SKZ14] that defined, for the static *stochastic block model* (SBM), a parametrization independent of $c_{\text{in}} - c_{\text{out}}$ to perform SC. This parametrization consists in choosing $(\xi, h) = (\gamma_d, \eta)$, where $\gamma_d = \alpha_c(T, \eta)/\sqrt{c\Phi}$. Let us first confirm that $B_{\gamma_d,\eta}$ indeed has isolated informative eigenvalues down to the detectability threshold.

Exploiting the result of Claim 9.2, the matrix $B_{\gamma_d,\eta}$ has an eigenvector correlated to the class labels equal to $\lambda_{\text{info},1} = \lambda_1^{\downarrow|\cdot|}(M_T(\sqrt{c\Phi\gamma\gamma_d}, \eta))$. First note that, by definition, $\sqrt{c\Phi\gamma_d^2} = \alpha_c(T, \eta)$, while $\sqrt{c\Phi\gamma^2} = \alpha$. For $\alpha > \alpha_c(T, \eta)$, then $\gamma > \gamma_d$, and, consequently $\sqrt{c\Phi\gamma\gamma_d} > \alpha_c(T, \eta)$. We then conclude that $\lambda_{\text{info},1} > 1$. From Claim 9.2, we further have that the radius of the bulk spectrum of $B_{\gamma_d,\eta}$ is equal to $L_{\gamma_d,\eta} = 1$. As such, the informative eigenvalue $\lambda_{\text{info},1}$ of $B_{\gamma_d,\eta}$ exists as soon as $\alpha > \alpha_c(T, \eta)$.

We now exploit Claim 3.3, that allows us to conclude that, since $L_{\gamma_d,\eta} = 1$, then the smallest eigenvalue of the bulk (*i.e.*, its left-edge) of $H_{\gamma_d,\eta}$ is asymptotically close to zero and all the eigenvectors associated to the isolated eigenvalues of $B_{\gamma_d,\eta}$. Among these, there is at least one (the one associated to $\lambda_{\text{info},1}$) that is informative.

These results are visually supported by Figure 9.8 and summarized with the following claim.

Claim 9.4 (Spectral clustering with $H_{\gamma_d,\eta}$)**.** *Let* $\gamma_d = \frac{\alpha_c(T,\eta)}{\sqrt{c\Phi}}$. *As* $n \to \infty$, *the following facts are verified with high probability*

- *the complex eigenvalues forming the bulk spectrum of* $B_{\lambda_d,\eta}$ *are asymptotically bounded within a disk in the complex plane of radius* $L_{\gamma_d,\eta} = 1$*:*
- *the smallest eigenvalues of the bulk spectrum of* $H_{\lambda_d,\eta}$ *tend to* 0^+*;*
- *the number of isolated negative eigenvalues of* $H_{\lambda_d,\eta}$ *is equal to the number of real isolated eigenvalues of* $B_{\lambda_d,\eta}$ *greater than* 1.

In particular, if $\alpha > \alpha_c(T,\eta)$, *at least one of the isolated real eigenvalues of* $B_{\lambda_d,\eta}$ *larger than* 1 *and one of the negative isolated eigenvalues of* $H_{\lambda_d,\eta}$ *are informative in the sense that their associated eigenvectors are correlated to the vector of community labels.*

Claim 9.4 indicates that, if $\alpha > \alpha_c(T,\eta)$, certainly there is one informative eigenvector (more precisely, mode 2 of Figure 9.5) which is associated with *one of the few isolated negative eigenvalues* of $H_{\gamma_d,\eta}$. Other informative eigenvectors (*e.g.* mode 4 of Figure 9.5) may be associated to negative eigenvalues of $H_{\gamma_d,\eta}$, but their existence is not guaranteed. The algorithmic advantage of Claim 9.4 with respect to Claim 9.3 is that, albeit the exact location of the informative eigenvectors is still unknown, for sure they are attached to negative eigenvalues of $H_{\gamma_d,\eta}$. We empirically confirm that using *all* the eigenvectors associated with the isolated negative eigenvalues (instead of only the desired informative eigenvector with unknown location) to form a low dimensional vector embedding of the nodes is redundant but it does not severely compromise the performance of the final *k-means* step. The choice $\xi = \gamma_d$ and $h = \eta$ therefore almost immediately induces an explicit algorithm applicable to arbitrary networks and which, as later discussed in Section 9.4, straightforwardly extends to graphs with $k > 2$ communities. Before detailing the algorithmic implementation of a SC algorithm based on $H_{\gamma_d,\eta}$, let us formulate the following remark.

Remark 9.4 (Comparison with the static Bethe-Hessian of [SKZ14])**.** *Recalling that the parameter* γ *behaves as the inverse of* r *appearing in the static Bethe-Hessian* H_r, *it is easily shown that the parametrization* $\xi = \gamma_d$ *is a generalization to the dynamical case of the choice* $r = \sqrt{c\Phi}$ *proposed in [SKZ14] for CD which is recovered for* $T = 1$ *(or equivalently for* $\eta = 0$*) for which we have that* $\alpha_c(T,\eta) = 1$. *While* γ_d *allows one to devise a well defined SC algorithm, like in the static case, the choice* $\xi = \gamma_d$ *is suboptimal and does not properly keep into account for the degree heterogeneity of the graph.*

With these results at hand, we are now in position to consider how Claim 9.4 can be translated into a practical algorithm that will then be tested on synthetic graphs generated from the DDCSBM.

Algorithm 9.1 : Community detection in sparse dynamical graphs

Input : Adjacency matrices $\{A^{(t)}\}_{t=1,\dots,T}$ of the undirected graphs $\{\mathcal{G}_t\}_{t=1,\dots,T}$; label persistence, η; number of clusters k.
Output : Estimated label vector $\hat{\boldsymbol{\ell}} \in \{1,\dots,k\}^{nT}$

begin
  Compute: $d_i^{(t)} \leftarrow \sum_{j=1}^{n} A_{ij}^{(t)}$, $c \leftarrow \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n} d_i^{(t)}$, $\Phi \leftarrow \frac{1}{nTc^2}\sum_{t=1}^{T}\sum_{i=1}^{n}\left(d_i^{(t)}\right)^2$;
  Compute: $\alpha_c(T,\eta)$ as in Claim 9.1, $\gamma_d \leftarrow \frac{\alpha_c(T,\eta)}{\sqrt{c\Phi}}$;
  Stack the κ eigenvectors of $H_{\gamma_d,\eta}$ with negative eigenvalues in the columns of $X \in \mathbb{R}^{nT\times\kappa}$;
  Normalize the rows of $X_{i,\bullet} \leftarrow X_{i,\bullet}/\|X_{i,\bullet}\|$;
  for $t = 1 : T$ **do**
    Estimate the community labels $\{\hat{\ell}_{i_t}\}_{i=1,\dots n}$ using k-class *k-means* on the rows $\{X_{i_t}\}_{i=1,\dots,n}$
  end
  return $\hat{\boldsymbol{\ell}} \in \{1,\dots,k\}^{nT}$.
end

9.4 ALGORITHM AND PERFORMANCE COMPARISON

9.4.1 ALGORITHM IMPLEMENTATION

We have previously summarized the main ideas behind a dynamical version of spectral clustering based on $H_{\gamma_d,\eta}$. These form the core of Algorithm 9.1. Yet, in order to devise a practical algorithm, applicable to a broad range of dynamical graphs, some aspects that go beyond the DDCSBM assumption should be taken into account.

So far, we dealt with $k = 2$ equal-size communities for which the DDCSBM threshold is well defined. Real networks may of course have multiple classes with asymmetrical sizes. As in the static case, we argue that, under Assumption 5.1, the left edge of the bulk spectrum of $H_{\gamma_d,\eta}$ is still asymptotically close to zero and that some of the eigenvectors associated with the isolated negative eigenvalues carry information for community reconstruction.[6] The value k is, in practice, also likely unknown. This also does not affect the idea of the algorithm which exploits all eigenvectors associated to the negative eigenvalues of $H_{\gamma_d,\eta}$, without the need of knowing k. The very choice of k is only required by *k-means* in the last step of SC and may be performed using off-the-shelf *k-means* compliant tools, *e.g.*, the *silhouettes method* [Rou87].

6 In passing, while $\alpha_c(T,\eta)$ is well defined regardless of k, for $k > 2$ classes of arbitrary size its value no longer corresponds to the position of a detectability threshold.

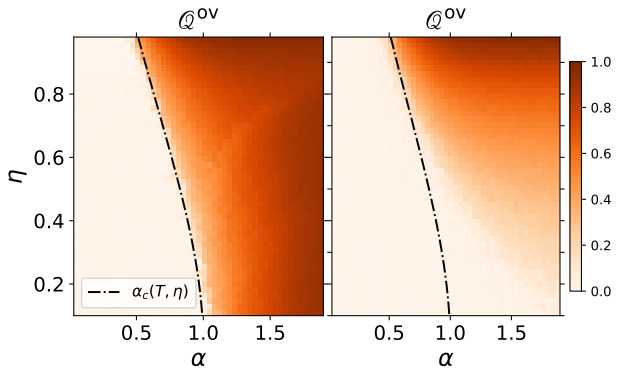


Figure 9.9: Overlap comparison at $t = T$ for Algorithm 9.1 (on the left) vs. [KV20] (on the right), in color gradient, for various detectability hardness levels α (x-axis) and label persistence η (y-axis); $n = 10\,000$, $T = 5$, $c = 10$, $\boldsymbol{\theta} = \mathbf{1}_n$; averaged over 4 samples.

A further important remark is that η is an input of Algorithm 9.1. If unknown, as it would in general be, one may choose an arbitrary $h \in [0,1)$ and $\xi = \alpha_c(T,h)$, to then perform spectral clustering on $H_{\xi,h}$: the leftmost edge of the bulk spectrum of $H_{\xi,h}$ is asymptotically close to zero for all h and consequently Algorithm 9.1 can be used in the same form. However, for a mismatched h, the detectability threshold now occurs beyond the optimal $\alpha_c(T,\eta)$. Close to the transition, this mismatch would give rise to fewer informative isolated negative eigenvalues than expected, resulting in a poor quality label assignment. As a workaround, one may browse through a discrete set of values for h and extract the h maximizing some quality measure, such as the resulting clustering *modularity*.

We further underline that in step 5, Algorithm 9.1 performs the hypersphere normalization step. Although not fully justified from a theoretical standpoint, the empirical evidence showed in Part II that this step helps to significantly improve the efficiency of the *k-means* step, correcting to some extent the deleterious contribution of an arbitrary degree distribution.

From a computational point of view, the bottleneck of Algorithm 9.1 is to compute the embedding X. The number of negative eigenvalues κ is not *a priori* known and only suspected to be in the interval $\{k,\dots,kT\}$. The strategy based on polynomial approximation and random projections detailed in Section 7.1.1 is here particularly adapted, allows us to estimate κ with $O(cnT)$ operations and to compute X with $O(ncTk^2)$ operations in the best case scenario and $O(ncT^4k^3)$ operation in the worst case. To give an order of magnitude, a simulation of Algorithm 9.1 for $n = 10^5$, $T = 5$ (resp., $n = 5\,000$, $T = 100$), $k = 2$, $c = 6$, $\eta = 0.5$, $\Phi = 1.6$, $\alpha = 2\alpha_c(T,\eta)$ takes on average approximately 1 minute (resp., 40 minutes) with its `CoDeBetHe.jl` implementation.

The number κ of negative eigenvalues to be computed, generally depends on T

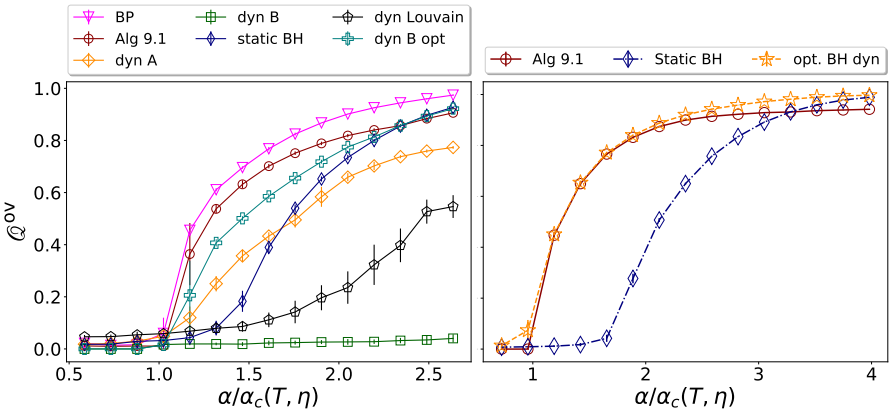


Figure 9.10: **Left**: mean overlap across all values of t, as a function of α, for Algorithm 9.1, BP [Gha+16], the dynamic adjacency matrix of [KV20] (dyn A), the dynamical non-backtracking of [Gha+16] (dyn B and dyn B opt), the static Bethe-Hessian of [DCT19] (static BH) and the dynamical Louvain algorithm of [Muc+10] (dyn Louvain); $n = 5\,000$, $T = 4$, $c = 6$, $\eta = 0.7$, $\boldsymbol{\theta} = \mathbf{1}_n$; averaged over 20 samples (3 for BP). For all plots, $k = 2$. **Right**: overlap comparison of Algorithm 9.1, the static Bethe-Hessian of [DCT20a] (StaticBH) and the reconstruction obtained using the eigenvector with zero eigenvalue of $H_{\gamma,\eta}$ (opt. BHdyn). For this simulation $n = 25\,000$, $k = 2$, $\boldsymbol{\theta} = \mathbf{1}_n$, $c = 6$, $T = 4$, $\eta = 0.9$. Averages are taken over 10 samples.

9.4.2 PERFORMANCE COMPARISON ON SYNTHETIC DATASETS

Algorithm 9.1 performs DCD for any value of η

Figure 9.9 compares the overlap as a function of α and η for Algorithm 9.1 versus the adjacency averaging method of [KV20] (which we recall assumes $\eta = 1 - o_n(1)$). The overlap is only considered at $t = T$ so to compare Algorithm 9.1 on even grounds with [KV20] which only outputs one partition (rather than one for every t). The theoretical detectability threshold line $\alpha = \alpha_c(T, \eta)$ visually confirms the ability of Algorithm 9.1 to assign non trivial class labels as soon as theoretically possible, as opposed to the method of [KV20] which severely fails at small values of η.

The left plot of Figure 9.10 then compares the average overlap performance of Algorithm 9.1 against competing methods, for varying detection complexities $\alpha/\alpha_c(T, \eta)$. Algorithm 9.1 is outperformed only by the BP algorithm,[7] but has a sensibly reduced computational cost. As we underlined already with Remark 9.3, for the non-backtracking method of [Gha+16] ("dyn B"), the authors suggest to use (as we did here) the eigenvector associated to the second largest eigenvalue of $B_{\gamma,\eta}$, which, as $H_{\gamma,\eta}$, may also have informative and uninformative eigenvalues in reversed order. The curve "dyn B opt" shows the performance obtained using all the isolated eigenvectors of $B_{\gamma,\eta}$ and it confirms, in agreement with Claim 9.3, that $B_{\gamma,\eta}$ can indeed

7 The codes used to obtain the BP performance displayed in Figure 9.10 are courtesy of Amir Ghasemian, author of [Gha+16].

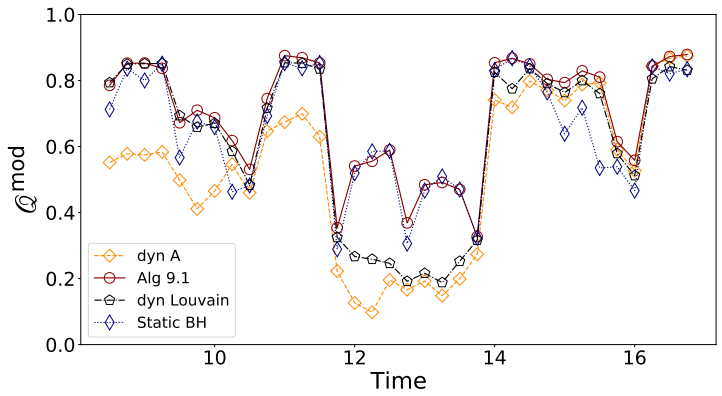


Figure 9.11: Modularity as a function of time for Algorithm 9.1 for $\eta = 0.55$, the dynamic adjacency matrix of [KV20] (dyn A), the dynamic Louvain algorithm [Muc+10] (dyn Louvain) and the static Bethe-Hessian of [DCT19] (static BH). The graph $\{\mathcal{G}_t\}_{t=1,\dots,T}$ is obtained from the *Primary school* network [GBC14; Ste+11] dataset, as in Section 9.4.3. For Algorithm 9.1, [DCT19] and [KV20], $k = 10$ is imposed.

make non-trivial community reconstruction for all $\alpha > \alpha_c(T, \eta)$. Note that, as in the static case [Krz+13; SKZ14], $B_{\gamma,\eta}$ is outperformed by $H_{\gamma_d,\eta}$ which, additionally, is symmetric and smaller in size, is well defined regardless of γ and is, therefore, a more suitable candidate for CD.

Interestingly, for large values of α, Algorithm 9.1 is slightly outperformed by the static Bethe-Hessian of [DCT19], independently run at each time-step. As discussed at the end of Section 9.3, the choice $\xi = \gamma_d$ is sub-optimal compared to the optimal (but out-of-reach in practice) choice $\xi = \gamma$, the difference becoming more visible as α increases away from α_c. Supposing one has access to an oracle for γ, running Algorithm 9.1 on $H_{\gamma,\eta}$ outperforms systematically the static Bethe-Hessian, as shown in the right plot of Figure 9.10. From a dynamical viewpoint, also, the large α regime is of least importance as a static algorithm can, alone, output a perfect reconstruction.

For completeness, Appendix C.3 provides further numerical performance comparison tests for different values of η, Φ, for $k > 2$, for larger values of n and T, and for graphs with clusters of different average sizes.

9.4.3 TEST ON SOCIOPATTERNS *PRIMARY SCHOOL*

This section shows the results of our experiments on the *Primary school* network [GBC14; Ste+11] of the `SocioPatterns` project. The dataset contains a temporal series of contacts between children and teachers of ten classes of a primary school. For each time $1 \leq t \leq T$, $\mathcal{G}_t$ is obtained considering all interactions from time t to time $t + 15$ min, starting from $t_1 =$ 8:30 am until $t_T = 5$ pm for $T = 34$. Figure 9.11 compares the modularity as a function of time for different clustering techniques. We empirically observe that, for this dataset, multiple values of η give similar results: this is

not surprising because the clusters are here well delineated and we are in the (less interesting) easy detection regime. The value $\eta = 0.55$ is considered as an input of Algorithm 9.1, because it approximately matches the value of η estimated from the inferred label vector $\hat{\ell}$ (see Equation (9.1)).

Figure 9.11 shows that Algorithm 9.1 performs better than [KV20; Muc+10] at all times, with a drastic gain during the lunch break, in which the community structure is harder to delineate. As compared to the static Bethe-Hessian, Algorithm 9.1 is slightly outperformed only on some times during the lunch break, while for other times it benefits from the label correlation. Defining a unique, time independent η certainly hampers the performance on this specific dataset in which a large η is expected during the lesson times, while a small η may be more appropriate during the lunch break.

9.5 CONCLUSION

In this chapter we tailored a novel spectral algorithm for CD on dynamical graphs. The proposed Algorithm 9.1 is capable of reconstructing communities as soon as theoretically possible, thereby largely outperforming state-of-the-art competing spectral approaches (especially when classes have a short-term persistence) while only marginally under-performing the optimal but more intensive BP algorithm.

A delicate feature of Algorithm 9.1 concerns the estimation of the class-persistence parameter η, if not available. We hinted in Section 9.4 at a greedy line-search solution which is however computationally inefficient and lacks of a sound theoretical support. This needs be addressed for Algorithm 9.1 to be more self-contained and applicable to real networks.

Beyond this technical detail, the present analysis only scratches the surface of dynamical community detection: the problem in itself is vast and many degrees of freedom have not been here accounted for. The label persistence η and community strength matrix C (and thus the parameter γ in a symmetric two-class setting) are likely to evolve with time as well. We empirically observed that Algorithm 9.1 naturally extends to this setting, each temporal block of the matrix $H_{(\cdot,\cdot)}$ now using its corresponding $\gamma_d^{(t)}$ and η_t. Yet, while Algorithm 9.1 seems resilient to a more advanced dynamical framework, it requires a much more involved theoretical description.

Nonetheless, this chapter showed that the results of Part II can be generalized to more involved settings and that the steps needed to devise an efficient SC algorithms are well delineated and can be retraced for more general clustering problems than CD on static, undirected graphs.

10

NISHIMORI MEETS BETHE: SPARSIFICATION OF KERNEL SPECTRAL CLUSTERING

Abstract

This chapter extends our work on SC *to the weighted setting. In particular a new relation between the* Nishimori temperature *parametrizing a distribution* P *and the* Bethe *free energy on random Erdős-Rényi graphs with edge weights distributed according to* P *is unveiled. Estimating the Nishimori temperature being a task of major importance in Bayesian inference problems, as a practical corollary of this new relation, a numerical method is proposed to accurately estimate the Nishimori temperature from the eigenvalues of the* Bethe Hessian *matrix of the weighted graph. The practical application of this result is in the i.i.d. sparsification of kernel matrices for cost efficient* spectral clustering *(*SC*) of high dimensional vectors.*

Spectral clustering (SC) techniques can be adopted as a practical *correlation clustering* algorithm, taking high-dimensional *feature* vectors [BBC04] as an input. This is done by first defining a *kernel matrix* representation that encodes the pairwise similarity between the elements of the input dataset and then extracting few relevant eigenvectors that provide a small dimensional representation of the initial dataset.

This problem has concrete applications to image, sound, or sentence classification [Lan+16] but, as we mentioned in the Introduction chapter of this manuscript, one of the main pitfalls of SC to categorize high dimensional vectors is its computational complexity and sampling techniques are often adopted to decrease the computational cost of SC [TL20].

In this chapter we consider a simple generative model to study the correlation clustering problem on kernel matrices with very few non-zero entries, allowing one to drastically reduce the computational complexity of SC. We then show the explicit relation between the considered generative model and the *random bond Ising model* (RBIM) in statistical physics. Our main result is a conjecture relating the smallest eigenvalue of the Bethe-Hessian matrix and the *Nishimori* temperature [Nis81] appearing in the physics of spin glasses. This relation allows us to devise an effective algorithm for the sparsified (hence cost efficient) version of SC.

10.1 CORRELATION CLUSTERING

10.1.1 INTRODUCTION

Let $\{\boldsymbol{z}_i\}_{i=1,\dots,n}$ be an n-vector dataset with $\boldsymbol{z}_i \in \mathbb{R}^{d_{\text{feat}}}$. These vectors represent discriminating *features* of some two-class data (say images) to be clustered in a fully unsupervised manner. In typical modern machine learning, d_{feat} is of the order of a few thousands for images and a few hundreds for natural language text representations, and it is not rare to try and classify up to millions of data vectors $\boldsymbol{z}_i$. The most elementary way to classify the vectors $\{\boldsymbol{z}_i\}_{i=1,\dots,n}$ is simply to directly run *k-means* algorithm in the d_{feat}-dimensional feature space. *K-means* is however known to fail for large d_{feat} [KKZ09] due to the fact that distances tend to concentrate in high dimensionality, a phenomenon that goes under the name of *curse of dimensionality* [Bel15]. The top line of Figure 10.1 provides a visualization of this phenomenon. To obtain this plot, we consider a label vector $\boldsymbol{\ell} \in \{1,2\}^n$ and draw n Gaussian vectors $\{\boldsymbol{z}_i\}_{i=1,\dots,n}$ with covariance matrix $I_{d_{\text{feat}}}$ and mean $\boldsymbol{\mu}_{\ell_i}$. On the left we show a scatter plot between two arbitrary components of the vectors $\{\boldsymbol{z}_i\}_{i=1,\dots,n}$, denoting with a different colour code the entries related to either class. This plot clearly evidences the effect of the *curse of dimensionality*. Furthermore, letting $Z \in \mathbb{R}^{n\times d_{\text{feat}}}$ be the matrix containing in its rows the vectors $\{\boldsymbol{z}_i\}_{i=1,\dots,n}$, on the right of the first row we plot the

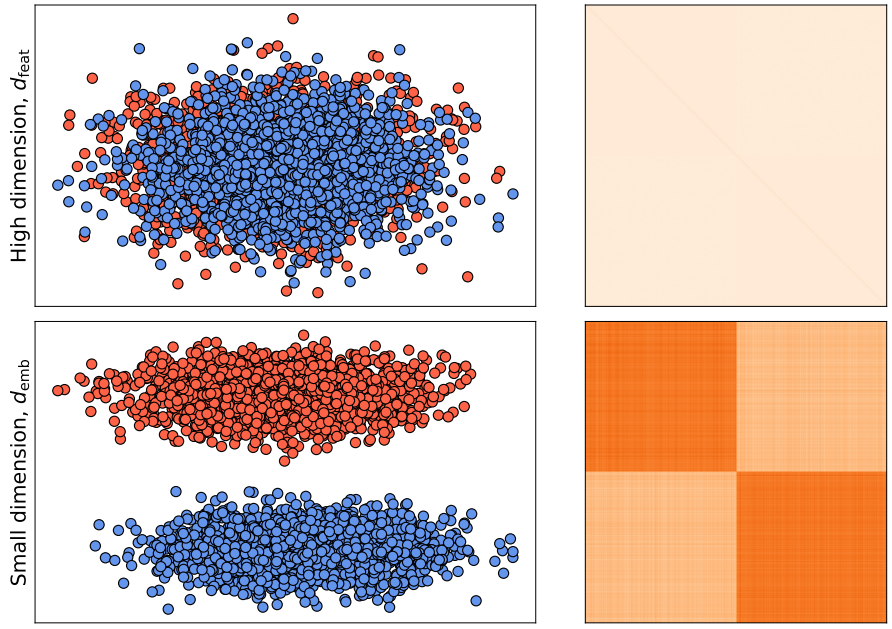


Figure 10.1: Example of dimensionality reduction using the covariance matrix $M_Z = \frac{1}{d_{\text{feat}}} ZZ^T$. Two Gaussian isotropic blobs of $n = 4\,000$ vectors in $d_{\text{feat}} = 3\,000$ with different means and identity covariance are considered. **Top**: on the left a scatter plot of two arbitrary chosen features in which the colour code shows the original label assignment; on the left the corresponding covariance matrix. **Bottom**: on the left the characteristic features after dimensionality reduction using the two leading eigenvectors of M_Z; on the right the corresponding covariance matrix.

covariance matrix[1] $M_Z = \frac{1}{d_{\text{feat}}} ZZ^T$ which just appears as a white square because all its entries are approximately equal.

A classical workaround is to perform a *dimensionality reduction* of the input vectors, embedding them in a lower dimensional space. A good embedding has the role to map "similar" vectors in high dimensions to points that are close in the embedded space. The analogy with SC for CD is clear: in that context, "similar" nodes correspond to nodes in the same community.

Pairwise measurements can be encoded with weighted edges of a graph

The most popular technique exploits a spectral approach [Ham+04; Sau+06; Ghoo6]: one starts by defining a kernel matrix $K(\{\boldsymbol{z}\}) \in \mathbb{R}^{n \times n}$, the entry $K_{ij}(\{\boldsymbol{z}\})$ of which evaluates some *affinity metric* between $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$; a collection of eigenvectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ for k typically equal (or close) to the number of classes are extracted and stacked in the columns of the matrix $X \in \mathbb{R}^{n \times k}$; the rows of X form the small dimensional embedding on which *k-means* can be safely run. A popular affinity function is merely the correlation $K(\{\boldsymbol{z}\}) = \frac{1}{d_{\text{feat}}} ZZ^T$, which is the choice adopted for *kernel principal*

1 Note the the entry ij of the covariance matrix is

$$(M_Z)_{ij} = \frac{1}{d_{\text{feat}}} \boldsymbol{z}_i^T \boldsymbol{z}_j = \frac{1}{2d_{\text{feat}}} [\|\boldsymbol{z}_i\|^2 + \|\boldsymbol{z}_i\|^2 - \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2] = \boldsymbol{\mu} + \frac{1}{2d_{\text{feat}}} \|\boldsymbol{z}_i - \boldsymbol{z}_j\|^2 + o_{d_{\text{feat}}}(1)$$

and is therefore directly related to the euclidean distance between the points in $\mathbb{R}^{d_{\text{feat}}}$.

component analysis [Pea01], but other choices exist, such as the *heat* kernel $K_{ij}(\{z\}) = \exp(-\|z_i - z_j\|^2/2\nu^2)$ for some $\nu > 0$, for instance. A visual example of how dimensionality reduction allows one to cluster high dimensional vectors is shown in the bottom row of Figure 10.1.

Sparse kernel matrices improve the computational complexity of SC

For large dimensional datasets, the computational complexity of building $K(\{z\})$ may become prohibitive: for d_{feat}, n beyond a few thousands, the $O(d_{\text{feat}}n^2)$ cost of building $K(\{z\})$ added to the (at least) $O(n^2)$ cost of computing the leading eigenvectors, makes SC hardly achievable on a modern home computer. To drastically decrease the computational complexity one may proceed to a two-level sparsification as recently proposed in [Zar+20; CCB21]: by randomly discarding elements of the d_{feat}-dimensional features z_i and by randomly dropping a number of evaluations of the correlations $K_{ij}(\{z\})$. This operation of course impedes the clustering performance, but, as surprisingly proved in [Zar+20; CCB21] under a "still rather dense graph" regime, the performance loss is negligible for a wide range of sparsity levels. To this end, let $S \in \{0,1\}^{n\times p}$ and $H \in \{0,1\}^{n\times n}$ (symmetric) be Bernoulli masks with parameters $\sqrt{\kappa/d_{\text{feat}}}$ and c/n, respectively.[2] The resulting sparsified kernel matrix then becomes

$$\tilde{J} = K(\{\tilde{z}\}) \circ H, \quad \text{where} \quad \tilde{z}_{i,l} = z_i S_{i,l}. \tag{10.1}$$

i.e., each entry of each of the feature vectors z_i is kept only with probability $\sqrt{\kappa/d_{\text{feat}}}$, while each measurement $K_{ij}(\{\tilde{z}\})$ is only performed with probability c/n. The computational complexity to build $\tilde{J}$ is thus scaled down to $O(\kappa cn)$. As just mentioned, in [Zar+20; CCB21] the authors still have to consider (for mathematical convenience) a rather dense regime in which c grows sufficiently fast with n and in practice do not solve the computational complexity problem.

Our goal is to devise a SC algorithm that is capable to operate in the sparse regime ($c = O_n(1)$), with computational complexity linear in n. To do so, we observe that the matrix $\tilde{J}$ of Equation (10.1) can be interpreted as a weighted adjacency matrix of a graph. In fact, every ij so that $\tilde{J}_{ij} = 0$ corresponds to a missing measurement or, in graph language, to a missing edge. On the opposite, when $\tilde{J}_{ij} \neq 0$, then it represents the weight of the edge (ij). This allows us to immediately rephrase the problem of sparsification of kernel matrices in the graph-theoretic language adopted in the previous chapters.

Following the same methodological pattern adopted so far, we now propose a generative model for $\tilde{J}$ on which our theoretical results will be based. First, we introduce an intermediate matrix J, generated as follows.

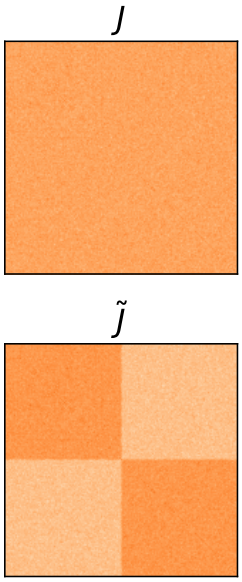
Definition 10.1 (Generative model of J)**.** *Let $\mathcal{G}(\mathcal{V},\mathcal{E})$ be a realization of a Erdős-Rényi graph of size n (Definition 1.9) with expected average degree c. Further let $J \in \mathbb{R}^{n\times n}$ be a weighted adjacency matrix on $\mathcal{G}(\mathcal{V},\mathcal{E})$ and distributed according to the following generative model. For all edges $(ij) \in \mathcal{E}$*

2 The choice of c is not a coincidence: H will enforce an average number of measurements per node (in graph-theory language, a degree) of c to the resulting graph.

the entries J_{ij} are generated independently (up to symmetry), for some $\beta_N > 0$, according to the distribution P

$$P(x) = p_0(|x|)\, e^{\beta_N x}, \tag{10.2}$$

where $p_0(\cdot)$ is an arbitrary non-negative function satisfying the normalization condition $\int_{-\infty}^{\infty} dx\; p_0(|x|)e^{\beta_N x} = 1$. If $(ij) \notin \mathcal{E}$, then $J_{ij} = 0$.



Note that, although not very customary when dealing with weighted graphs, according to Definition 10.1, J_{ij} may be negative. We called J an "intermediate" matrix because it is helpful to define a generative model for $\tilde{J}$. Suppose that the vector $\boldsymbol{\ell} \in \{1,2\}^n$ assigns a label to each node in $\mathcal{V}$. Define $\boldsymbol{\sigma} \in \{\pm 1\}^n$ so that $\sigma_i = 1$ if $\ell_i = 1$ and $\sigma_i = -1$ otherwise. Let J be generated as per Definition 10.1, then the matrix $\tilde{J}$ is obtained as follows

$$\tilde{J} = J \circ \boldsymbol{\sigma}\boldsymbol{\sigma}^T. \tag{10.3}$$

We now detail the main ingredients motivating the considered generative model for the matrix $\tilde{J}$.

- The expectation of $\tilde{J}$: from Definition 10.1, combined with (10.3), the expectation of a non-zero entry of $\tilde{J}$ reads $\mathbb{E}[\tilde{J}_{ij}] = \mathbb{E}[J_{ij}]\sigma_i\sigma_j$. The expectation $\mathbb{E}[J_{ij}]$ is the same for all $(ij) \in \mathcal{E}$ and it is positive, in fact

$$\mathbb{E}[x] = \int_{-\infty}^{\infty} dx\; p_0(|x|)e^{\beta_N x}\, x = \int_0^{\infty} dx\; \underbrace{p_0(|x|)\, 2\mathrm{sh}(\beta_N x)\, x}_{>0},$$

  consequently, we immediately see that sign of $\mathbb{E}[\tilde{J}_{ij}]$ is positive for i, j in the same class and it is negative otherwise. This is a reasonable requirement that the matrix $\tilde{J}$ defined in Equation (10.1) should satisfy and it implies that, *on average*, $K_{ij}(\{\boldsymbol{z}\})$ is larger when $\ell_i = \ell_j$. This is readily verified in the Gaussian example considered in Figure 10.1 for $\boldsymbol{\mu}_1 = -\boldsymbol{\mu}_2$ in which, letting $K_{ij}(\{\boldsymbol{z}\}) = \frac{1}{d_{\text{feat}}}\boldsymbol{z}_i^T\boldsymbol{z}_j$, then $\mathbb{E}[\tilde{J}_{ij}] \propto \sigma_i\sigma_j$. It must be noted that the matrix $\tilde{J}$ of Equation (10.1) must satisfy $\tilde{J}\mathbf{1}_n = \mathbf{0}_n$ to be reasonably approximated by $\tilde{J}$ as in Equation (10.3). This can however be easily obtained shifting all the input vectors $\{\tilde{\boldsymbol{z}}_i\}_{i=1,\dots,n} = \{\boldsymbol{z}_i - \boldsymbol{f}\}_{i=1,\dots,n}$, where $\boldsymbol{f} \in \mathbb{R}^{d_{\text{feat}}}$ is an i-independent vector.

  Further note that, since the function $p_0(|x|)$ needs not to be specified, the considered generative model includes a large number of random matrices satisfying the condition $\mathbb{E}[\tilde{J}_{ij}] = \sigma_i\sigma_j$.

- The noise in $\tilde{J}$: the parameter β_N (that is the main focus of the remainder of the chapter) controls the noise in the realization of $\tilde{J}$. In fact, considering the limit $\beta_N \to 0$, we get $P(x) \to p_0(|x|)$ and the sign of J_{ij} (and consequently of $\tilde{J}_{ij}$) is essentially random. On the opposite, when $\beta_N \to \infty$, $J_{ij} > 0$ with high probability and consequently $\mathrm{sign}(\tilde{J}_{ij}) = \sigma_i\sigma_j$. Tuning β_N between these two extremes, one determines the likelihood that $\mathrm{sign}(J_{ij}) = \sigma_i\sigma_j$: the smaller this likelihood is, the more relevant is the contribution of the noise, allowing the considered generative model to recreate hard detection tasks.

- The independence hypothesis: a major concern comparing the matrix $\tilde{J}$ defined as per Equation (10.1) and (10.3) is certainly the independence of the entries assumed in Definition 10.1. In fact, by construction, the entries $K_{ij}(\{z\}), K_{ik}(\{z\})$ and $K_{jk}(\{z\})$ are far from being independent. Independence however asymptotically holds between the non-zero entries of $\tilde{J}$ for a sufficient level of sparsification as a consequence of the absence of short cycles in the underlying Erdős-Rényi graph. In the example we gave, if $(ij), (ik) \in \mathcal{E}$, then $\mathbb{P}[(jk) \in \mathcal{E}] = o_n(1)$, hence the dependence of $K_{jk}(\{z\})$ on $K_{ij}(\{z\}), K_{ik}(\{z\})$ can be neglected.

For these reasons, we claim that devising a SC algorithm to infer the vector σ from a realization of $\tilde{J}$ allows one to cluster the input vectors $\{z_i\}_{i=1,\dots,n}$ into their genuine classes. We now formalize the problem from a Bayesian perspective, describing its clear relation with statistical physics.

10.1.2 FROM BAYESIAN INFERENCE TO STATISTICAL PHYSICS

Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be the realization[3] of an Erdős-Rényi graph whose nodes are divided in two non-overlapping classes, labelled via the vector $\sigma \in \{\pm 1\}^n$ and let $\tilde{J} \in \mathbb{R}^{n \times n}$ be the matrix generated according to Equation (10.3). Then the following relation holds:

$$\mathbb{P}(\tilde{J}|\sigma) = \prod_{(ij)\in\mathcal{E}} p_0(|\tilde{J}_{ij}|) e^{\beta_{\mathrm{N}} \tilde{J}_{ij}\sigma_i\sigma_j}.$$

In order to infer the vector σ from a realization of $\tilde{J}$ we may express the posterior probability distribution as follows:

$$\mathbb{P}(\sigma|\tilde{J}) = \frac{\mathbb{P}(\tilde{J}|\sigma)\mathbb{P}(\sigma)}{\mathbb{P}(\tilde{J})} = \frac{1}{Z_{\beta_{\mathrm{N}},\tilde{J}}} \exp\left\{ \sum_{(ij)\in\mathcal{E}} \beta_{\mathrm{N}} \tilde{J}_{ij}\sigma_i\sigma_j \right\}. \tag{10.4}$$

This is the fundamental relation of Bayes optimal inference. Note that computing the marginals of $\mathbb{P}(\sigma|\tilde{J})$ is equivalent to determine the magnetization of an Ising model on $\tilde{J}$ at β_{N} (see Chapter 2), later referred to as *Nishimori temperature* [Nis81].

The relation between β_{N} and the Ising model has a long history [Nis81] in the physics of disordered systems [BY86] that raised the scientific interest even before its relation with Bayes inference was unveiled [Iba99]. To explain the importance of the Nishimori temperature in statistical physics, let us introduce the *random bond Ising model* (RBIM). Let $\mathcal{H}_J(s)$ be the Ising Hamiltonian (Equation (2.2)) associated to a weighted graph J generated according to Definition 10.1 and denote with $\mu(s)$ the Boltzmann distribution:

3 The graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is thus *fixed* and not a random variable.

$$\mathcal{H}_J(\boldsymbol{s}) = -\sum_{(ij)\in\mathcal{E}} J_{ij}s_i s_j, \tag{10.5}$$

$$\mu_\beta(\boldsymbol{s}) = \frac{1}{Z_{\beta,J}} e^{-\beta\mathcal{H}_J(\boldsymbol{s})}. \tag{10.6}$$

Equations (10.5, 10.6) define the RBIM in which, we recall, β_{N} is a parameter of the generative model of J. Nishimori noticed in [Nis81] that at $\beta = \beta_{\mathrm{N}}$, *i.e.*, when the temperature of the system coincides with the Nishimori temperature, the exact expression of $\mathbb{E}\left[\langle\mathcal{H}_J(\boldsymbol{s})\rangle_\beta\right]$ can be computed with elementary mathematical tools, where $\langle\cdot\rangle_\beta$ denotes the averaging over the Boltzmann distribution (10.6) while $\mathbb{E}[\cdot]$ is the averaging over the distribution of Equation (10.2). It has also been shown [NS01; ZK16, for instance] that the RBIM at the Nishimori temperature is either in the *ferromagnetic* configuration (in which $\langle s_i\rangle_\beta > 0$ for all i) or in the *paramagnetic* configuration (for which $\langle s_i\rangle_\beta = 0$ for all i). In particular, the system is never in the *spin-glass phase* under which local order of $\boldsymbol{s}$ appears despite there being no global magnetization. These relevant properties drew research attention to this particular temperature [Geo+85; GRL01; TPV09] since [Nis81].

Relating the Bayes optimal solution (10.4) with the RBIM (10.6) is trivially done introducing a *gauge transformation*. If fact, let $\tilde{s}_i = s_i\sigma_i$ and recall that $\tilde{J}_{ij} = J_{ij}\sigma_i\sigma_j$, then

$$\mathcal{H}_J(\boldsymbol{s}) = -\sum_{(ij)\in\mathcal{E}} J_{ij}s_i s_j = -\sum_{(ij)\in\mathcal{E}} \tilde{J}_{ij}\sigma_i\sigma_j\tilde{s}_i\sigma_i\tilde{s}_j\sigma_j = -\sum_{(ij)\in\mathcal{E}} \tilde{J}_{ij}\tilde{s}_i\tilde{s}_j,$$

where we exploited $\sigma_i^2 = 1$ for all i. This allows one to create an explicit mapping between the posterior distribution of $\boldsymbol{\sigma}$ and the distribution of $\tilde{\boldsymbol{s}}$ and to use the powerful tools and theoretical results borrowed from statistical physics to devise an efficient algorithm to perform clustering. In the Bayes optimal inference, however, the value of β_{N} must be known, an assumption that for unsupervised clustering tasks is often unrealistic. Earlier works have resorted to studying the problem of mismatched inference, *i.e.* inference performed for $\beta \neq \beta_{\mathrm{N}}$ [ZK16].

10.1.3 OUR CONTRIBUTION: RELATING NISHIMORI TO BETHE

Our main result consists in going beyond mismatched inference by providing an efficient estimate of the Nishimori temperature, β_{N}. To this end, we first draw an explicit relation between β_{N} and the smallest eigenvalue of the (weighted) Bethe-Hessian matrix; this relation holds under the previously introduced setting, so in particular for a matrix J supported over a (possibly sparse) Erdős-Rényi graph $\mathcal{G}(\mathcal{V},\mathcal{E})$. Besides, we observe and argue that, although the Bethe approximation is particularly adapted to sparse (locally

tree-like) graphs $\mathcal{G}(\mathcal{V},\mathcal{E})$, the *Nishimori-Bethe relation* holds for any degree of sparsity (that is, even when $\mathcal{G}(\mathcal{V},\mathcal{E})$ does not behave locally as a tree).

The main consequences of the Nishimori-Bethe relation, and our main contributions, consist in

- the design of a new efficient spectral algorithm which estimates β_{N} with asymptotically perfect accuracy; the algorithm is based on an iterative fast search of a well-parametrized Bethe-Hessian matrix with the smallest amplitude eigenvalue close to zero;
- a new spectral algorithm to approximately solve the Bayesian node classification inference problem of Equation (10.4), which outperforms commonly deployed state-of-the-art alternatives. We in particular claim that this spectral algorithm is capable of performing non trivial inference as soon as the Bayes optimal solution can;
- although we claim that these algorithms are still valid under dense graphs $\mathcal{G}(\mathcal{V},\mathcal{E})$, they are specifically adapted to the sparse regime where $c = O_n(1)$; this practically allows for small computational and memory storage costs when performing SC on large datasets; we specifically support this fact by a concrete application to the classification of 40 000 high resolution images using our proposed sparse but extremely efficient spectral algorithm.

Given the trivial relation between the Bayes optimal inference problem and studying the magnetization of the RBIM at $\beta = \beta_{\mathrm{N}}$, we chose, for simplicity, to work with the matrix J rather than $\tilde{J}$. Several relations will however be made to relate the physics and inference results.

Let us now proceed with some relevant properties of the Nishimori temperature in the RBIM the will be of utmost importance in the remainder.

10.2 BASIC PROPERTIES OF THE RANDOM BOND ISING MODEL

10.2.1 PHASE DIAGRAM

Consider a realization of a Erdős-Rényi graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ with expected average degree c. Let $J \in \mathbb{R}^{n\times n}$ be a weighted adjacency matrix associated to $\mathcal{G}(\mathcal{V},\mathcal{E})$ and generated as per Definition 10.1. We want to study the phase diagram, *i.e.* the behavior of $\langle \boldsymbol{s} \rangle_\beta$, for $\boldsymbol{s}$ a random variable distributed according to Equations (10.5,10.6). Figure 10.2 displays this phase diagram for the $\pm$J model which corresponds to the particular choice of $p_0(|x|)$ leading to

$$P(x) = p\delta(x - \mathrm{J}_0) + (1-p)\delta(x + \mathrm{J}_0), \quad \text{for } p \in [1/2, 1], \quad \mathrm{J}_0 \in \mathbb{R}^+,$$

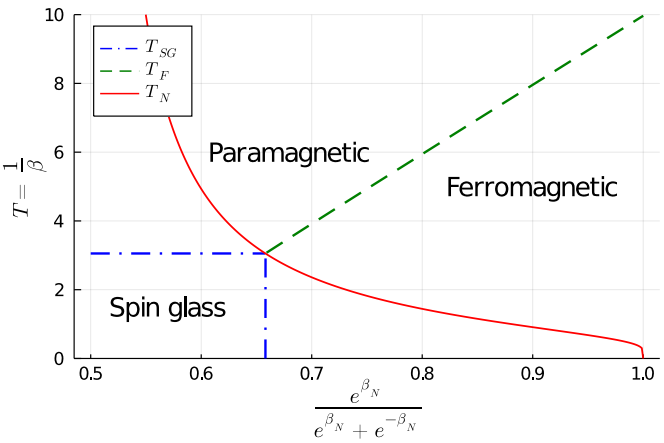


Figure 10.2: Phase diagram of the RBIM for $J_{ij} \in \{-1, 1\}$. The x axis goes from $\frac{1}{2}$ for $\beta_N = 0$ to 1 for $\beta_N \to \infty$. The y axis represents T, the inverse of β. The dashed green line is the inverse of β_F, the dash dotted blue line is the inverse of β_{SG} and the solid red line is the inverse of β_N.

but qualitatively similar diagram can be obtained for different choices of $p_0(|x|)$, like the Edwards-Anderson model, for instance [EA75] in which

$$P(x) = \frac{1}{\sqrt{2\pi\nu^2}} \exp\left\{-\frac{(x - J_0)^2}{2\nu^2}\right\}.$$

To best understand the phase diagram in Figure 10.2, first consider the role played by the two parameters β and β_N. For increasing values of β_N, there is a larger probability for each edge J_{ij} to carry a positive weight and the minimum of $\mathcal{H}_J(\boldsymbol{s})$ is achieved for $\boldsymbol{s} \approx \mathbf{1}_n$. For small values of β_N, instead, multiple local minima appear. Concerning β, instead, for small values, the Boltzmann distribution (Equation (10.6)) tends towards a uniform distribution, while, for large values, the configurations with small energy $\mathcal{H}_J(\boldsymbol{s})$ have a larger probability. Consequently, for large β and β_N the average configuration of $\boldsymbol{s}$ tends to align towards $\mathbf{1}_n$: this corresponds to the *ferromagnetic* configuration. Conversely, for small values of β_N, several edges carry a negative weight, introducing *frustration* in the system that is found in the *spin-glass* phase, for which local order of the spins may be observed ($\frac{1}{n}\sum_i \langle s_i \rangle_\beta^2 \neq 0$), but globally the magnetization is null ($\frac{1}{n}\sum_i \langle s_i \rangle_\beta = 0$). Finally, at large values of β, the system is in the *paramagnetic* phase, for which the spins are randomly aligned and $\langle \boldsymbol{s} \rangle_\beta = \mathbf{0}_n$.

In the particular case where $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is a sparse Erdős-Rényi random graph, the Bethe approximation (2.3.2) predicts the asymptotically exact position of the transitions between the three phases: the *paramagnetic-ferromagnetic* transition occurs at $\beta = \beta_F$ and the *paramagnetic-spin glass* transition occurs at $\beta = \beta_{SG}$, also known as the de Almeida-Thouless transition [Tho86]. We recall from Section 3.4 that the values of β_F (resp. β_{SG}) is determined imposing the leading eigenvalue (resp. the radius of the bulk) of

the weighted non-backtracking with weights equal to $\text{th}(\beta J_e)$ equals one. Exploiting Theorem 3.4 [SM20], this leads to:

$$c \cdot \mathbb{E}[\text{th}(\beta_{\text{F}} J_{ij})] := 1 \tag{10.7}$$

$$c \cdot \mathbb{E}[\text{th}^2(\beta_{\text{SG}} J_{ij})] := 1. \tag{10.8}$$

Given these premises, we now discuss some relevant properties valid on the Nishimori line, *i.e.* when $\beta = \beta_{\text{N}}$.

10.2.2 RELEVANT PROPERTIES AT THE NISHIMORI TEMPERATURE

First of all, let us introduce the *quenched internal energy density*, defined as $u(\beta) := \frac{1}{n}\mathbb{E}[\langle \mathcal{H}_J(\boldsymbol{s})\rangle_\beta]$, where we recall that $\langle\cdot\rangle_\beta$ denotes an average taken over the Boltzmann distribution (10.6) and $\mathbb{E}[\cdot]$ is the average over the distribution of Equation (10.2). It was shown in [Nis81] that $u(\beta_{\text{N}})$ takes a particularly simple expression:

Exact expression of the quenched internal energy at the Nishimori temperature

$$\begin{aligned} u(\beta_{\text{N}}) &= \frac{1}{n}\mathbb{E}[\langle \mathcal{H}_J(\boldsymbol{s})\rangle_{\beta_{\text{N}}}] = -\frac{1}{n}\sum_{(ij)\in\mathcal{E}} \mathbb{E}\left[J_{ij}\langle s_i s_j\rangle_{\beta_{\text{N}}}\right] \\ &= -\frac{1}{n}\sum_{(ij)\in\mathcal{E}} \mathbb{E}[J_{ij}\,\text{th}(\beta_{\text{N}} J_{ij})]. \end{aligned}$$

The first two equalities are true by definition. The elegance of the result of [Nis81] lies in the last relation that identifies – inside the expectation $\mathbb{E}[\cdot]$ – the term $\langle s_i s_j\rangle_{\beta_{\text{N}}}$ with $\text{th}(\beta_N J_{ij})$. We recall from Section 2.3.2, for β sufficiently small, the system is in the paramagnetic phase $\langle \boldsymbol{s}\rangle_\beta = 0$ and, under the Bethe approximation, the relation $\langle s_i s_j\rangle_\beta = \text{th}(\beta J_{ij})$ is verified for any underlying β_{N}. This informally introduces a relation between the Bethe free energy at the paramagnetic point and the Nishimori temperature, which is at the centre of Claim 10.1.

We introduce the following property of the probability distribution of Equation (10.2). This relation will be of fundamental use in the following and, in passing, allows us to rewrite $u(\beta_{\text{N}})$ as in [Nis81].

Property 10.1. *Let $f(x)$ be an arbitrary odd function. Then*

$$\mathbb{E}[f(x)\cdot \text{th}(\beta_{\text{N}} x)] = \mathbb{E}[f(x)].$$

The proof is easily obtained by straightforward calculation. As a consequence of Property 10.1, the quenched internal energy density at the Nishimori temperature takes the simple expression:

$$u(\beta_{\text{N}}) = -\frac{1}{n}\sum_{(ij)\in\mathcal{E}} \mathbb{E}[J_{ij}] = -\frac{\bar{d}}{2}\cdot\mathbb{E}[J_{ij}],$$

where $\bar{d}$ denotes the average node degree in the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$.

Secondly, we recall a well celebrated property of the Nishimori temperature, which states the absence of *replica symmetry breaking* on the Nishimori line [NS01; ZK16] or, equivalently, that the RBIM at the Nishimori temperature is never in the spin glass phase. This result can be visually understood in Figure 10.2 by noticing that the Nishimori temperature is either in the paramagnetic or ferromagnetic phase. Moreover, exploiting Property 10.1 and the definitions of β_F, β_{SG} in Equations (10.7, 10.8), on an Erdős-Rényi graph one finds that $\beta_{SG} = \beta_N \iff \beta_F = \beta_N$. Consequently, there exists a tricritical point where $\beta_F = \beta_{SG} = \beta_N$.

$$\mathbb{E}[\text{th}^2(\beta_N J)] = \mathbb{E}[\text{th}(\beta_N J)]$$

Recalling the connection with statistical inference problems, such as inferring σ in Equation (10.4), first note that β_N is the Bayes optimal inference temperature in the sense that there exists no other β that can asymptotically achieve better inference performance and, therefore, if inference cannot be performed at $\beta = \beta_N$, then it is theoretically infeasible. This occurs when the marginals of Equation (10.4) asymptotically give equal probabilities for each σ_i to take either values ± 1. In terms of the phase diagram, this corresponds to being in the *paramagnetic* phase, so that $\beta_N < \beta_F$. In order for non-trivial reconstruction to be possible, the condition $\beta_N < \beta_{SG} < \beta_F$ must be imposed and it corresponds to the *detectability threshold* of this problem [Saa+16]. When the condition is met, the system is in the *informative* configuration in which each *spin* gets oriented towards its planted value σ_i. This being said, replacing (or, erroneously estimating) β_N by $\beta \neq \beta_N$ in Equation (10.4), it may occur that, even though inference is theoretically possible (as $\beta_N < \beta_{SG} < \beta_F$), the estimated labels $\hat{\sigma}$ for the mismatched β are not aligned with the ground truth σ. This never happens at $\beta = \beta_N$ for which inference is achieved *as soon as theoretically possible*.

Detectability threshold

With this short introduction on the Nishimori temperature at hand, we now present our main result which relates β_N to the spectrum of the non-backtracking and Bethe-Hessian matrices of the underlying graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

10.3 A RELATION BETWEEN β_N AND THE BETHE FREE ENERGY

This section introduces our main theoretical result, which draws a connection between the Nishimori temperature and the Bethe-Hessian and non-backtracking matrices associated to the Hamiltonian (10.5). Our main result, precisely consists in (i) a claim on the location of the eigenvalues of the weighed *non-backtracking* matrix and, as a result of the claim; (ii) an explicit relation between the underlying Nishimori temperature and a specific eigenvalue of the *non-backtracking* matrix; (iii) a claim relating the smallest eigenvalue of the Bethe-Hessian matrix and β_N.

Moreover, Section 10.3.3 elaborates an algorithm to estimate β_N, which finds significant importance in statistical inference problems.

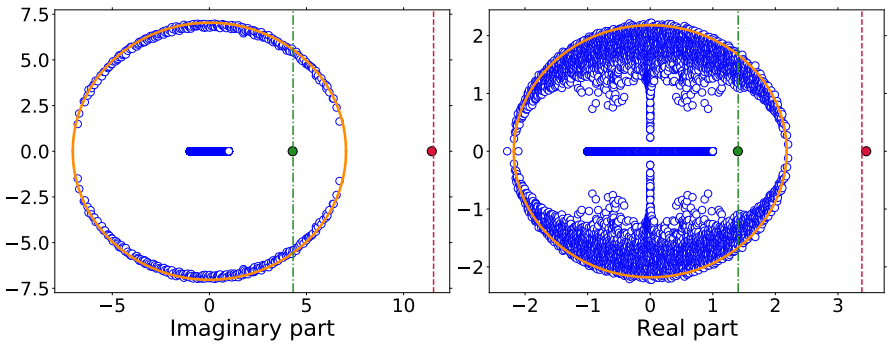


Figure 10.3: Spectrum of the matrix B_ω in the complex plane. The entries J_{ij} are generated independently according to $\mathcal{N}(J_0, \nu^2)$. The edge weights are defined as $\omega_{ij} = \mathrm{th}(\beta J_{ij})$. **Left**: dense regime, $n = 250$, $c = 2\log^2(n)$, $J_0 = 1$, $\nu = 4$, $\beta = 1$. **Right**: sparse regime, $n = 3\,000$, $c = 5$, $J_0 = 1, \nu = 1, \beta = 10$. **For both plots**, the dashed red line corresponds to $c\mathbb{E}[\mathrm{th}(\beta J)]$, the dash-dotted green line to $\mathbb{E}[\mathrm{th}^2(\beta J)]/\mathbb{E}[\mathrm{th}(\beta J)]$, while the orange continuous line is the circle in the complex plane centered at the origin and of radius $\sqrt{c\mathbb{E}[\mathrm{th}^2(\beta J)]}$.

10.3.1 MAIN RESULT

The main mathematical object of our central claim is the weighted non-backtracking matrix $B_\omega \in \mathbb{R}^{|\mathcal{E}_\mathrm{d}|\times|\mathcal{E}_\mathrm{d}|}$, defined as

$$B_\omega = BU_\omega,$$

where we recall that $\mathcal{E}_\mathrm{d}$ is the set of directed edges of $\mathcal{G}(\mathcal{V},\mathcal{E})$, $B \in \{0,1\}^{|\mathcal{E}_\mathrm{d}|\times|\mathcal{E}_\mathrm{d}|}$ is the (unweighted) non-backtracking matrix of the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ (Definition 1.8), $U = \mathrm{diag}(\boldsymbol{\omega})$ and $\boldsymbol{\omega} \in \mathbb{R}^{|\mathcal{E}_\mathrm{d}|}$ encodes the edge weights.

We now proceed to studying the spectrum of the matrix B_ω in the case where $\mathcal{G}(\mathcal{V},\mathcal{E})$ is a Erdős-Rényi graph and its weights ω_e are drawn i.i.d. satisfying $|\omega_e| < 1$ with $\mathbb{E}[\omega] > 0$ sufficiently large. The interest of this setting in relation to the RBIM and the Nishimori temperature is to consider $\omega_e = \mathrm{th}(\beta J_e)$ for $\beta_\mathrm{N} > \beta_\mathrm{SG}$ and J as per Definition 10.1. In this particular case, one of the eigenvalues of B_ω has a direct relation with β_N.

Our central result concerns the existence of a real isolated eigenvalue in the spectrum of B_ω and is summarized with Claim 10.1.

Claim 10.1. *(Spectrum of B_ω) Let $\mathcal{G}(\mathcal{V},\mathcal{E})$ be a realization of an Erdős-Rényi random graph with n nodes ($n \to \infty$) and expected average degree c. For each undirected edge $(ij) \in \mathcal{E}$ a weight $\omega_{ij} = \omega_{ji} \in (-1,1)$ is assigned independently at random. Further assume that $\mathbb{E}[\omega_{ij}^2]/\mathbb{E}[\omega_{ij}] \geq 1$ and $\mathbb{E}[\omega_{ij}^2]/\mathbb{E}^2[\omega_{ij}] < c$. Then, with high probability, the spectrum of B_ω can be described as follows:*

- *there exist only two real eigenvalues in the spectrum of B_ω with modulus greater or equal to one:*

$$\lambda_1 = c\mathbb{E}[\omega] + o_n(c)\,, \quad \lambda_{-1} = \frac{\mathbb{E}[\omega^2]}{\mathbb{E}[\omega]} + o_n(1). \tag{10.9}$$

The eigenvalue λ_1 is the largest in modulus in the spectrum of B_ω;

- *all eigenvalues with non-zero imaginary part have a modulus bounded by $L_\omega = \sqrt{c\mathbb{E}[\omega^2]} + o_n(\sqrt{c})$.*

Note that Claim 10.1 does not make the assumption that $c \to \infty$ as $n \to \infty$, nor that $c = O_n(1)$. Extensive simulations indeed concur in suggesting that the claim holds in both dense and sparse graph regimes. The claim is thus stated for *any* average degree, so long that the underlying graph is of the Erdős-Rényi type. In detail, the technical condition $\mathbb{E}[\omega_{ij}^2]/\mathbb{E}^2[\omega_{ij}] < c$ is set to enforce that the leading eigenvalue λ_1 is greater than the radius of the bulk spectrum (hence that it is isolated) and that λ_{-1} is smaller than the radius of the bulk: a transition occurs at $\mathbb{E}[\omega_{ij}^2]/\mathbb{E}^2[\omega_{ij}] = c$ where both eigenvalues coincide: $\lambda_1 = \lambda_{-1}$. This inequality condition will thus ensure, when it comes to statistical inference, that non-trivial $\sigma \in \{\pm 1\}^n$ configurations can be theoretically recovered (*i.e.*, that the inference problem is feasible). As a practical support to Claim 10.1, Figure 10.3 displays the spectrum of the matrix B_ω in both moderately dense ($c \sim \log^2(n)$) and sparse ($c = O_n(1)$) regimes.

The fundamental corollary of Claim 10.1 is that, in the case of present interest where $\omega_e = \mathrm{th}(\beta J_e)$, from Equation (10.9), the *inner* eigenvalue λ_{-1} of B_ω is equal to

$$\lambda_{-1} = \frac{\mathbb{E}[\mathrm{th}^2(\beta J)]}{\mathbb{E}[\mathrm{th}(\beta J)]} + o_n(1).$$

Exploiting Property 10.1, it follows immediately that, at $\beta = \beta_N$,

At $\beta = \beta_N$ the inner isolated eigenvalue of B equals 1

$$\lambda_{-1} \underset{\beta=\beta_N}{=} 1 + o_n(1).$$

Tuning the value of β until $\lambda_{-1} = 1$ thus provides *a method to estimate* β_N. The question on how to efficiently exploit this essential remark from an algorithmic standpoint will be further discussed in Section 10.3.3. Before pushing further our main line of deductions, let us provide some arguments in support of Claim 10.1, treating separately the dense and sparse regimes.

10.3.2 SUPPORTING ARGUMENTS

Dense graphs

We first consider a dense graph regime, *i.e.*, when the average degree c goes to infinity faster than $\log(n)$. This argument is inspired from the proof provided in [CZ20] for unweighted dense graphs with a community structure. The proof of [CZ20] can be straightforwardly adapted to the *binary* case in which $\omega_e \in \{\pm\omega\}$, but does not unfold so directly for $\omega_e \in \mathbb{R}$.

On dense Erdős-Rényi graphs

$\|D - cI_n\| = o_n(c)$

We recall that the main advantage of the dense regime follows from the fact that the degree distribution of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is almost regular and the Erdős-Rényi graph is close to a c-regular graph, the analysis of which is easier to handle. This makes it possible to relate the eigenvalues of $B_{\boldsymbol{\omega}}$ to those of $W \in \mathbb{R}^{n\times n}$, defined as $W_{ij} = \omega_{ij}$ if $(ij) \in \mathcal{E}$ and zero otherwise. The idea is to create a sequence of matrices $M(\boldsymbol{g}) \in \mathbb{R}^{2n\times 2n}$ (one for each eigenvector $\boldsymbol{g}$ of $B_{\boldsymbol{\omega}}$), in the spirit of a proof proposed by Bass of the celebrated Ihara-Bass formula [HST06], and to show that all the eigenvalues of $M(\boldsymbol{g})$ can be approximated, in the large n limit, by the eigenvalues of a common matrix M_0 independent of $\boldsymbol{g}$, so long that $\boldsymbol{g}$ is an eigenvector corresponding to an eigenvalue λ of $B_{\boldsymbol{\omega}}$ for which $|\lambda| \geq 1$. It is the precise study of the spectrum of the limiting M_0 which induces the results of Claim 10.1.

More specifically, let $\boldsymbol{g} \in \mathbb{C}^{|\mathcal{E}_{\mathrm{d}}|}$ be an eigenvector of $B_{\boldsymbol{\omega}}$ with eigenvalue λ, satisfying $|\lambda| \geq 1$ and let $\boldsymbol{\omega} \in \mathbb{R}^{|\mathcal{E}_{\mathrm{d}}|}$ be the vector containing the weights of the edges of $\mathcal{G}(\mathcal{V}, \mathcal{E})$ (and recall that $\omega_{ij} = \omega_{ji}$). Then define the vectors $\boldsymbol{\psi}(\boldsymbol{g}), \tilde{\boldsymbol{\psi}}(\boldsymbol{g}) \in \mathbb{C}^n$ as

$$\psi_i(\boldsymbol{g}) = \sum_{j\in\partial i} \omega_{ij} g_{ij}\,, \qquad \tilde{\psi}_i(\boldsymbol{g}) = \sum_{j\in\partial i} \omega_{ij}^2 g_{ji} \tag{10.10}$$

and $F(\boldsymbol{g}) \in \mathbb{C}^{n\times n}$ be any matrix satisfying

$$\big[F(\boldsymbol{g})\boldsymbol{\psi}(\boldsymbol{g})\big]_i = \sum_{j\in\partial i} \omega_{ij}^3 g_{ij}. \tag{10.11}$$

We now wish to relate the quantities $\boldsymbol{\psi}(\boldsymbol{g}), \tilde{\boldsymbol{\psi}}(\boldsymbol{g}), F(\boldsymbol{g})$ to the eigenvalues of $B_{\boldsymbol{\omega}}$. In particular,

$$\begin{aligned}\lambda\psi_i(\boldsymbol{g}) = \psi_i(B_{\boldsymbol{\omega}}\boldsymbol{g}) &= \sum_{j\in\partial i} \omega_{ij} \sum_{(k\ell)} \delta_{jk}(1-\delta_{i\ell})\omega_{kl} g_{k\ell} \\ &= \sum_{j\in\partial i} \omega_{ij} \left[\sum_{l\in\partial j} \omega_{j\ell} g_{j\ell} - \omega_{ji} g_{ji}\right] = [W\boldsymbol{\psi}(\boldsymbol{g})]_i - \tilde{\psi}_i(\boldsymbol{g})\end{aligned}$$

and, similarly,

$$\begin{aligned}\lambda\tilde{\psi}_i(\boldsymbol{g}) = \tilde{\psi}_i(B_{\boldsymbol{\omega}}\boldsymbol{g}) &= \sum_{j\in\partial i} \omega_{ij}^2 \sum_{(k\ell)} \delta_{ik}(1-\delta_{j\ell})\omega_{k\ell} g_{k\ell} \\ &= \sum_{j\in\partial i} \omega_{ij}^2 \left[\sum_{\ell\in\partial i} \omega_{i\ell} g_{i\ell} - \omega_{ij} g_{ij}\right] = [D_W\boldsymbol{\psi}(\boldsymbol{g})]_i - [F(\boldsymbol{g})\boldsymbol{\psi}(\boldsymbol{g})]_i\,,\end{aligned}$$

where $D_W \in \mathbb{R}^{n\times n}$ is the diagonal matrix with $[D_W]_{ii} = \sum_{j\in\partial i} \omega_{ij}^2$. Thus, the eigenvalue λ is also an eigenvalue of the matrix

$$M(\boldsymbol{g}) = \begin{pmatrix} W & -I_n \\ D_W - F(\boldsymbol{g}) & 0 \end{pmatrix}. \tag{10.12}$$

The main difficulty of the analysis is of course introduced by the matrix $F(\boldsymbol{g})$ which is different for each eigenvector of $B_{\boldsymbol{\omega}}$ associated to $|\lambda| > 1$. In

the *binary* case in which $W_{ij} \in \{\pm\omega\}$ for all $(ij) \in \mathcal{E}$, this term simplifies: combining Equations (10.10) and (10.11), we get $F(\boldsymbol{g})\boldsymbol{\psi} = \omega^2\boldsymbol{\psi}$ and $F(\boldsymbol{g})$ thus simplifies *for all* $\boldsymbol{g}$ into $F(\boldsymbol{g}) = \omega^2 I_n$; this allows for a straightforward adaptation of the proof of [CZ20]. The non-binary case is, however, more involved, but the term $\big(D_W - F(\boldsymbol{g})\big)\boldsymbol{\psi}$ is still dominated by D_W:

F is negligible with respect to D_W

$$\left|\frac{[F(\boldsymbol{g})\boldsymbol{\psi}(\boldsymbol{g})]_i}{\psi_i(\boldsymbol{g})}\right| = \left|\frac{\sum_{j\in\partial i}\omega_{ij}^3 g_{ij}}{\sum_{j\in\partial i}\omega_{ij} g_{ij}}\right| = o_n(c).$$

For the last equality, we exploited the fact that ω_{ij} and ω_{ij}^3 are both bounded in $(-1,1)$ and have the same sign.[4] Consequently, $F(\boldsymbol{g})$ can be regarded as a small perturbation of D_W. Further exploiting the concentration of the degrees, one may thus write

$$\Big\|\Big(D_W - F(\boldsymbol{g})\Big) - c\mathbb{E}[\omega^2] I_n\Big\| = o_n(c).$$

The eigenvalues of $M(\boldsymbol{g})$ can therefore be approximated by those of

$$M_0 = \begin{pmatrix} W & -I_n \\ c\mathbb{E}[\omega^2] I_n & 0 \end{pmatrix}. \tag{10.13}$$

The spectrum of M_0 is trivially related to the spectrum of W. Letting $\{\lambda_i(W)\}_{i=1,\dots,n}$ be the eigenvalues of W and $\{\lambda_{0,i}\}_{i=\pm1,\dots,\pm n}$ those of M_0, by the block determinant formula [Siloo, Section 5], it comes that

$$\lambda_{0,\pm i} = \frac{\lambda_i(W) \pm \sqrt{\lambda_i(W)^2 - 4c\mathbb{E}[\omega^2]}}{2}.$$

In particular, it unfolds that

$$\lambda_i^2(W) \geq 4c\mathbb{E}[\omega^2] \quad\Longrightarrow\quad \lambda_{0,-i} = \frac{c\mathbb{E}[\omega^2]}{\lambda_{0,i}} \equiv \frac{L_\omega^2}{\lambda_{0,i}} \tag{10.14}$$

$$\lambda_i^2(W) < 4c\mathbb{E}[\omega^2] \quad\Longrightarrow\quad |\lambda_{0,\pm i}| = \sqrt{c\mathbb{E}[\omega^2]} \equiv L_{\boldsymbol{\omega}}. \tag{10.15}$$

Applying successively Wigner's semi-circle theorem [Wig58] and Bauer-Fike's theorem [BF60], we thus have that

$$\lambda_1(W) = c\mathbb{E}[\omega] + o_n(\sqrt{c}) \tag{10.16}$$

$$|\lambda_i(W)|_{i\geq 2} \leq \sqrt{c\mathbb{E}[\omega^2]} + o_n(\sqrt{c}). \tag{10.17}$$

Combining (10.14, 10.15) with (10.17, 10.16), along with the fact that the eigenvalues $\{\lambda_{0,\pm i}\}_{i=1,\dots,n}$ are a close approximation to the eigenvalues of $B_{\boldsymbol{\omega}}$ with modulus greater than one, we obtain the formulation of Claim 10.1. Figure 10.4 compares the spectra of the matrices $M(\boldsymbol{g})$ and M_0, which should be themselves compared to the left display in Figure 10.3.

4 This step is reasonable but non-rigorous, the main theoretical difficulty arising from the dependence between ω_{ij} and g_{ij}.

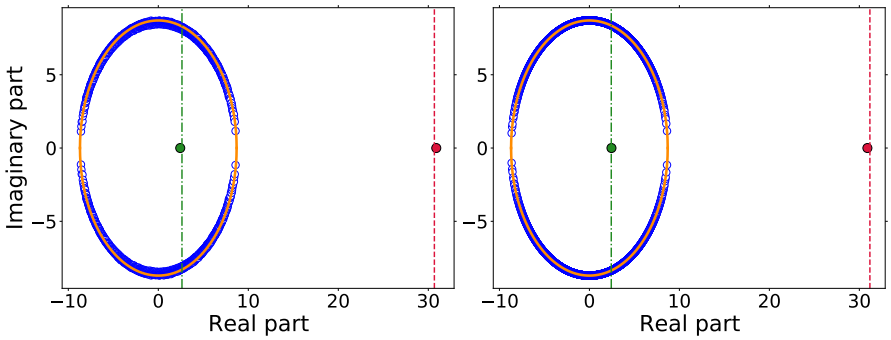


Figure 10.4: **Left**: spectrum of the matrix $M(\boldsymbol{g})$ defined in Equation (10.12) with $\boldsymbol{g}$, one of the eigenvectors of B_ω attached to a complex eigenvalue. The expression of $F(\boldsymbol{g})$ used in this simulation can be found in Appendix A of [DCT21] **Right**: Spectrum of M_0, defined in Equation (10.13). The graph considered is the same for the two matrices, with $n = 1\,500$, $c = \log^2(n)$. The matrix $W = \mathrm{th}(\beta J)$, with $\beta = 1$ and the entries J_{ij} are i.i.d. normal variable with $J_0 = 1$ and $\nu = 3$. The red dotted line is at $c\mathbb{E}[\mathrm{th}(\beta J)]$, the green dash-dotted line is the position of $\mathbb{E}[\mathrm{th}^2(\beta J)]/\mathbb{E}[\mathrm{th}(\beta J)]$, while the orange solid line is the circle in the complex plane of radius $\sqrt{c\mathbb{E}[\mathrm{th}^2(\beta J)]}$.

This technical argument provides important intuitions on the spectrum of B_ω: (i) the leading eigenvalue of B_ω (the largest in modulus) is determined by the expectation of the entries of W; (ii) the radius of the bulk of B_ω is determined by the expectation of the squared entries of W; (iii) all eigenvalues of B_ω with modulus greater than one come in pairs: they are complex conjugates if their imaginary part is non-zero or *harmonic conjugate* if they are real. This last observation justifies the existence of a real isolated eigenvalue *inside* the bulk of B_ω.

As a downside, the setting considered in this section is, somehow, too simplistic. The analysis allowed us to neglect the term $F(\boldsymbol{g})$, which plays the role of the *Onsager reaction term* [MPV87] which does not appear in the naïve mean-field approximation but plays a crucial role in the Bethe approximation. The fact that $F(\boldsymbol{g})$ can be neglected thus indicates that the regime under consideration is somehow *too simple*, the spectral behavior of B_ω being fully determined by W. Consequently, we next discuss the far more interesting *sparse* regime in which the reaction term plays a fundamental role. In the sparse regime, the structure of the spectrum of the matrix B_ω is essentially preserved, as well as the fact that all its eigenvalues all come in real harmonic or complex conjugate pairs.

Sparse graphs

We recall that the Bethe approximation is asymptotically exact on locally tree-like graphs, such as sparse Erdős-Rényi random graphs. In this case, the spectrum of W is no longer formed by an isolated, real eigenvalue and a

bulk of eigenvalues close to each other that follow the semi-circle law, as it happens in the *dense* regime discussed in the previous paragraph.

The non-backtracking matrix $B_{\boldsymbol{\omega}}$, instead, essentially preserves the same spectral structure as in the dense regime, in which the *bulk* eigenvalues are bounded by a circle in the complex plane, as shown in Figure 10.3 and confirmed by Theorem 3.4 of [SM20] in which, however, the authors do not mention the existence of *inner* real eigenvalues in the spectrum of $B_{\boldsymbol{\omega}}$. Yet, the position of the leading eigenvalue of $B_{\boldsymbol{\omega}}$ and the radius of its bulk spectrum are the same as in the *dense* graph case. We then conjecture, supported by extensive simulations, that also the inner isolated eigenvalue has the same position as in the dense regime, given by the square radius of the bulk, divided by the leading eigenvalue of $B_{\boldsymbol{\omega}}$.

Returning to the implications of Claim 10.1 of immediate interest, recall that the claim makes it possible to relate the Nishimori temperature to the specific eigenvalue λ_{-1} of the matrix $B_{\boldsymbol{\omega}}$. From a numerical standpoint though, λ_{-1} is not easily accessible since λ_{-1} is smaller in modulus than most of the complex eigenvalues of $B_{\boldsymbol{\omega}}$, while not being the smallest in modulus (see Figure 10.3), one needs to compute all the bulk eigenvalues of $B_{\boldsymbol{\omega}}$ in order to access λ_{-1}: this comes at an impractical computational cost of $O_n((cn)^3)$ with state of the art methods. We next show that, as a consequence of Claim 10.1, the weighted Bethe-Hessian matrix (Definition 2.2) can be efficiently used to estimate β_N in the RBIM with a computational cost scaling as $O_n(cn)$.

10.3.3 EFFICIENT ESTIMATION OF β_N

This section elaborates on our final relation between the Bethe-Hessian matrix and the Nishimori temperature, as well as on how the respective spectra can be related to the phase diagram of Figure 10.2.

First of all, let us recall the explicit expression of the weighted Bethe-Hessian $H_{\beta,J}$ that will be considered in the following

$$\left(H_{\beta,J}\right)_{ij} = \delta_{ij}\left(1 + \sum_{k\in\partial i}\frac{\tanh^2(\beta J_{ik})}{1-\tanh^2(\beta J_{ik})}\right) - \frac{\tanh(\beta J_{ij})}{1-\tanh^2(\beta J_{ij})}. \tag{10.18}$$

The weighted Bethe-Hessian matrix

We recall from Theorem 2.1 that whenever $\lambda = 1$ is an eigenvalue of $B_{\boldsymbol{\omega}}$ for $\omega_e = \tanh(\beta J_e)$, then $\det[H_{\beta,J}] = 0$. In order to best understand how $H_{\beta,J}$ can be used to efficiently estimate β_N, let us consider the behavior of its eigenvalues as a function of β. First of all, at sufficiently high temperature (small β), for all β_N the system is in the *paramagnetic* phase, for which $\boldsymbol{m} = \mathbf{0}_n$ is a minimum of the Bethe free energy. Consequently $H_{\beta,J}$ is positive definite, while all eigenvalues of $B_{\boldsymbol{\omega}}$ are smaller than one.

Consider now β_N to be sufficiently large, so that the system undergoes to a *paramagnetic-ferromagnetic* phase transition (see Figure 10.2). For $\beta = \beta_F$ defined as $c\mathbb{E}[\tanh(\beta_F J)] = 1$, the leading eigenvalue of $B_{\boldsymbol{\omega}}$ is (asymptotically) equal to 1 and one of the eigenvalues of $H_{\beta_F,J}$ is equal to zero. This eigenvalue is necessarily the smallest.

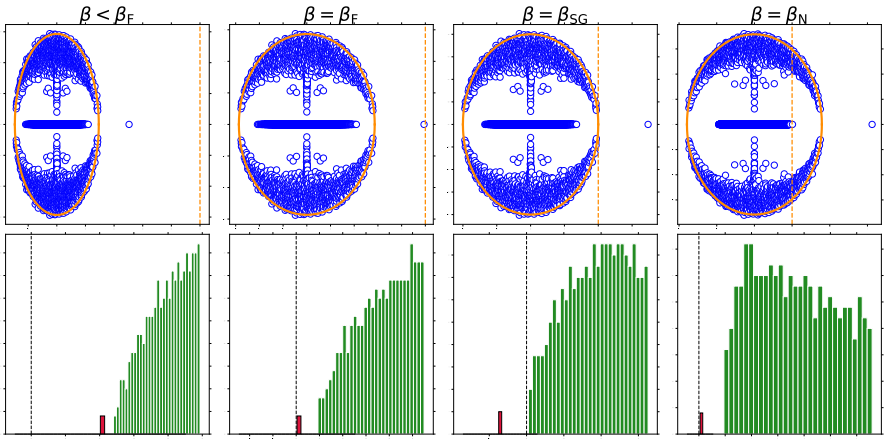


Figure 10.5: **First row**: spectrum of the matrix B_ω in the complex plane for different values of β; **Second row**: histogram of the eigenvalues of $H_{\beta,J}$ (zoomed in on the smallest eigenvalues) for different values of β. **First column**: $\beta = 0.5\beta_F$, paramagnetic phase; **Second column**: $\beta = \beta_F$ paramagnetic-ferromagnetic transition; **Third column**: $\beta = \beta_{SG}$ paramagnetic-spin glass phase transition; **Fourth column**: $\beta = \beta_N$, Nishimori temperature. For all matrices, the same graph was used with $n = 1\,000$, $c = 10$. The weights of the edges are $\omega_{ij} = \text{th}(\beta J_{ij})$ for the different values of β just described and the J_{ij} are drawn independently from a Gaussian distribution $\mathcal{N}(J_0, \nu)$, with $J_0 = 1$ and $\nu = 1.5$. The orange lines vertical line is at $x = 1$, while the black vertical line in the second row is at $x = 0$.

For small values of β_N, the system undergoes the *paramagnetic-spin glass* phase transition (see Figure 10.2) at the temperature $\beta = \beta_{SG}$ defined so that $c\mathbb{E}[\text{th}^2(\beta_{SG}J)] = 1$. Here, the radius of the bulk of B_ω is (asymptotically) equal to one and the bulk of $H_{\beta_{SG},J}$ is (asymptotically) equal to zero.

Finally, further decreasing the temperature, at $\beta = \beta_N$ defined by the relation $\mathbb{E}[\text{th}^2(\beta_N J)] = \mathbb{E}[\text{th}(\beta_N J)]$, the eigenvalue λ_{-1} is equal to one and the smallest eigenvalue of $H_{\beta_N,J}$ reaches zero for the second time. In Figure 10.5 we show the spectra of the matrices B_ω and $H_{\beta,J}$ at $\beta < \beta_F$, $\beta = \beta_F, \beta_{SG}, \beta_N$ that confirm the relation between the spectra of these two matrices and the phase diagram.

With these fundamental observations at hand, we now show how one can efficiently estimate β_N, exploiting the smallest eigenvalue of $H_{\beta,J}$. The proposed estimator of β_N reads

An efficient estimate for β_N

$$\hat{\beta}_N = \max_\beta \left\{ \beta \,:\, \lambda_1^{\uparrow}(H_{\beta,J}) = 0 \right\},$$

Under this definition, not only does $\hat{\beta}_N$ provide a consistent estimate of β_N for J distributed as Definition 10.1, but it also provides the "best guess" of an hypothetically corresponding β_N for matrices J which would follow a different distribution. Indeed, $\hat{\beta}_N$ has the advantage of always being defined, even for arbitrary matrices J, while having a clear interpretation for the

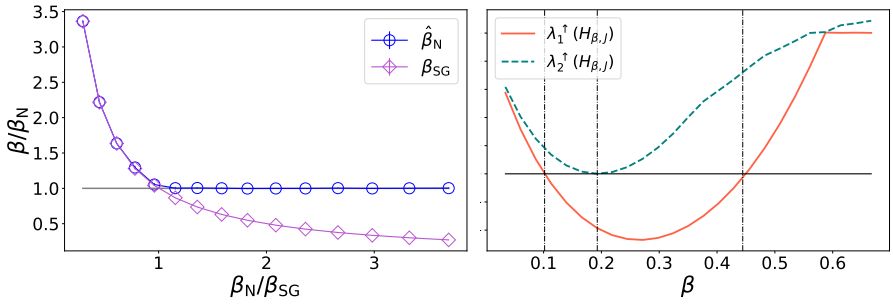


Figure 10.6: **Left**: computation of $\hat{\beta}_N$ for different values of β_N. The blue dots represent the ratio between $\hat{\beta}_N$, computed with Algorithm 10.1 and the analytical value of β_N. The purple diamonds are the value of β_{SG}/β_N, while the orange line is at $y = 1$. For these plots, $n = 10\,000$ and $c = 5$. The weights of the non-zero entries of J are distributed i.i.d. according to $\mathcal{N}(J_0, \nu^2)$ for J_0 ranging from $J_0 = 0.5$ to $J_0 = 4$ and $\nu = 3.5$. Averages are taken over ten samples. **Right**: behavior of the two smallest eigenvalues of $H_{\beta,J}$ as a function of β. The solid line indicates the smallest eigenvalue, while the dotted line is the second smallest. The vertical lines are set at $\beta_F < \beta_{SG} < \beta_N$. For this simulation, $n = 30\,000$ and $c = 10$. The weights of the matrix J are distributed i.i.d. according to a $\mathcal{N}(J_0, \nu^2)$ with $J_0 = 1$ and $\nu = 1.5$.

class of matrices that fall under Definition 10.1. This definition is particularly reminiscent of Algorithm 5.1 for CD.

The equation $\lambda_1^{\uparrow}(H_{\beta,J}) = 0$ has two solutions for $\beta \approx \beta_F$ and $\beta \approx \beta_N$ with $\beta_N > \beta_F$, justifying our proposed estimator. As a visual support, the right plot of Figure 10.6 shows $\lambda_{1,2}^{\uparrow}(H_{\beta,J})$ as a function of β.

The basic idea of the proposed algorithm to compute $\hat{\beta}_N$ precisely consists in starting from $\beta = \beta_{SG}$ to then find the value of $\beta > \beta_{SG}$ for which $\lambda_1^{\uparrow}(H_{\beta,J}) = 0$. Following this argument, we propose an iterative algorithm based on Courant-Fischer theorem (similar to Algorithm 5.1) to compute $\hat{\beta}_N$. The output of Algorithm 10.1 is depicted in the left display of Figure 10.6. Note in particular that, as long as $\beta_{SG} < \beta_N$, *i.e.*, as long as $\mathbb{E}[\text{th}^2(\beta J_{ij})]/\mathbb{E}^2[\text{th}(\beta J_{ij})] < c$, the value of $\hat{\beta}_N$ is a good estimate of β_N. When the condition is instead not met, $\hat{\beta}_N$ simply coincides with β_{SG}. The numerical advantage of exploiting the Bethe-Hessian matrix is decisive. First $H_{\beta,J}$ is *symmetric* and of size $n \times n$ *regardless of the average node degree*. Most importantly, the only eigenvalue of $H_{\beta,J}$ that needs be computed is the one of smallest amplitude, so that $\hat{\beta}_N$ can be estimated at a $O_n(nc)$ operations.

With these theoretical results laid out, we now move back to our initial problem of performing correlation clustering on high dimensional vectors and see how Claim 10.1 has a practical interest to accomplish this task.

Algorithm 10.1 : `Compute_`$\hat{\beta}_{\mathrm{N}}$

Input : Weighted adjacency matrix of a graph $J \in \mathbb{R}^{n\times n}$, precision error $\epsilon \in \mathbb{R}^{+}$.

Output : Value of $\hat{\beta}_{\mathrm{N}} \in \mathbb{R}^{+}$.

begin
- Compute c, the average degree of the underlying unweighted graph: $c = \frac{1}{n}\sum_i \sum_j \mathbb{I}(J_{ij} \neq 0)$;
- Compute $\hat{\beta}_{\mathrm{SG}}$ by solving $c\mathbb{E}[\mathrm{th}^2(\hat{\beta}_{\mathrm{SG}} J_{ij})] = 1$;
- Set $t = 1$ and $\beta_t \leftarrow \hat{\beta}_{\mathrm{SG}}$;
- Initialize $\delta \leftarrow +\infty$;
- **while** $\delta > \epsilon$ **do**
  - Compute $H_{\beta_t, J}$ (Equation (10.18)) ;
  - Compute $\lambda_1^{\uparrow}(H_{\beta_t,J})$ and its associated eigenvector $\boldsymbol{x}_t$;
  - Define the function $f_t(\beta') = \boldsymbol{x}_t^T H_{\beta',J} \boldsymbol{x}_t$, for $\beta' \in \mathbb{R}^{+}$;
  - Compute β_{t+1} by solving $f_t(\beta_{t+1}) = 0$;
  - Update $\delta \leftarrow |\lambda_1^{\uparrow}(H_{\beta_t,J})|$;
  - Increment $t \leftarrow t + 1$;
- **end**

end

return β_{t-1}.

10.4 ALGORITHM AND PERFORMANCE COMPARISON

10.4.1 RELAXATION OF THE BAYES OPTIMAL INFERENCE

In Section 10.1.2 we showed that the problem of Bayesian inference in a two class setting given a matrix $\tilde{J}$ as in Equation (10.3) can be easily mapped to the RBIM that we extensively studied in Section 10.3. This mapping is induced by a simple *gauge* transformation, that enables the use of Algorithm 10.1 to estimate β_{N} directly from $\tilde{J}$. Recall that $\tilde{J}_{ij} = J_{ij}\sigma_i\sigma_j$, where $\sigma_i \in \{\pm 1\}$ encodes the label information of node i. Let $\boldsymbol{x} \in \mathbb{R}^n$ be an eigenvector of $H_{\beta,\tilde{J}}$ with eigenvalue λ and let $\boldsymbol{y}$ have entries $y_i = x_i\sigma_i$. Then

$$\lambda y_i = \lambda x_i \sigma_i = \sigma_i \sum_j \left(H_{\beta,\tilde{J}}\right)_{ij} x_j = \sigma_i \sum_j \left(H_{\beta,J}\right)_{ij} \sigma_i\sigma_j x_j = \left(H_{\beta,J}\boldsymbol{y}\right)_i$$

so that λ is an eigenvalue of $H_{\beta,J}$ with eigenvector $\boldsymbol{y}$. Consequently, the smallest eigenvalue of $H_{\beta_{\mathrm{N}},\tilde{J}}$ is asymptotically close to zero and Algorithm 10.1 can indeed be used to estimate β_{N}.

Algorithm 10.2 : The Nishimori-Bethe relation for sparse SC

Input : Weighted adjacency matrix of a graph $\tilde{J} \in \mathbb{R}^{n\times n}$, precision error $\epsilon \in \mathbb{R}$.

Output : Value of $\hat{\beta}_{\mathrm{N}} \in \mathbb{R}^{+}$, estimated label vector $\hat{\boldsymbol{\sigma}} \in \{-1,1\}^{n}$.

begin

- Shift the non-zero $\tilde{J}_{ij}$ as: $\tilde{J}_{ij} \leftarrow \tilde{J}_{ij} - \frac{1}{2|\mathcal{E}|}\mathbf{1}_n^T \tilde{J}\mathbf{1}_n$;
- Compute $\hat{\beta}_{\mathrm{N}} \leftarrow$ `Compute_`$\hat{\beta}_{\mathrm{N}}$ (Algorithm 10.1) ;
- Compute $H_{\hat{\beta}_{\mathrm{N}},\tilde{J}}$ (Equation (10.18)) ;
- Compute $\boldsymbol{x} \leftarrow$ the eigenvector associated to $\lambda_1^{\uparrow}(H_{\hat{\beta}_{\mathrm{N}},\tilde{J}})$;
- Estimate $\hat{\boldsymbol{\sigma}}$ as the output of 2-class *k-means* on the entries of $\boldsymbol{x}$;

end

return β_t, $\hat{\boldsymbol{\sigma}}$.

From this trivial computation, it also appears that $\boldsymbol{y}$ is expected to be strongly related to the vector $\boldsymbol{\sigma}$. In fact, exploiting Property 10.1 of the Bethe-Hessian matrix associated to J, we can write at $\beta = \beta_{\mathrm{N}}$:

$$\mathbb{E}\left[H_{\beta_{\mathrm{N}},J}\right] = I_n + \mathbb{E}\left[\frac{\mathrm{th}(\beta J_{ij})}{1-\mathrm{th}^2(\beta J_{ij})}\right](D-A)\,.$$

From a straightforward calculation (see Proposition 1 of [VL07]), the vector $\mathbf{1}_n$ is the eigenvector of $\mathbb{E}[H_{\beta_{\mathrm{N}},J}]$ associated to its eigenvalue of smallest amplitude. As a consequence, from the relation between $\boldsymbol{x}$ and $\boldsymbol{y}$ (or equivalently between $\tilde{J}$ and J), the vector $\boldsymbol{\sigma}$ is the eigenvector of $\mathbb{E}[H_{\beta_{\mathrm{N}},\tilde{J}}]$ associated with its eigenvalue of smallest amplitude. Consequently, the eigenvector with zero eigenvalue of $H_{\beta_{\mathrm{N}},\tilde{J}}$ is a close approximation of $\boldsymbol{\sigma}$.

This conclusion immediately translates into Algorithm 10.2, a numerical method to infer the genuine node classification $\boldsymbol{\sigma}$. The performance of Algorithm 10.2 is tested on synthetic datasets and evaluated against two commonly deployed SC techniques that we here briefly review.

Competing spectral methods

A very classical spectral clustering method in weighted graphs exploits the *weighted Laplacian* matrix $L = \bar{D} - \tilde{J}$, where $\bar{D} = \mathrm{diag}(|\tilde{J}|\mathbf{1}_n)$. As shown in [VL07; Kun+10], the eigenvector attached to the smallest eigenvalue of L provides a (discrete to continuous) relaxed solution of the NP-hard optimization *signed ratio-cut* graph clustering problem which attempts to maximize the number of edges with positive weights connecting nodes in the same community, while minimizing the number of edges with negative weights connecting nodes in opposite communities. The relation of this method with the SC algorithm of [Fie73] for CD should be evident and similar comments to the ones made in Chapter 8 are valid in this context as well.

The second method[5] we are to consider instead performs SC on the leading eigenvectors of the matrix $\tilde{J}$. This is a very popular strategy to perform SC in

5 Note that in the context of correlation clustering and for the choice $K = \frac{1}{d_{\mathrm{emb}}} ZZ^T$, this is nothing but the *principal component analysis*.

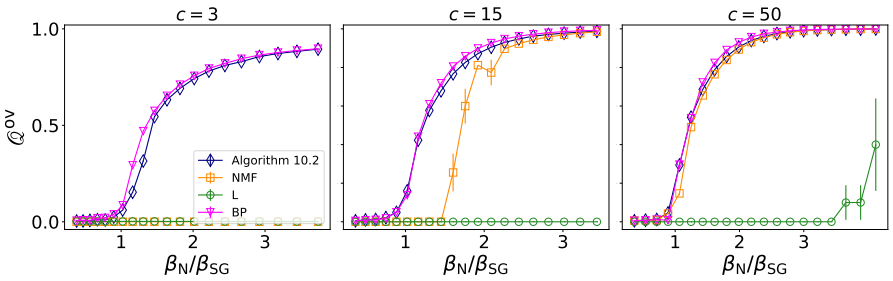


Figure 10.7: Overlap performance as a function of β_N/β_{SG} and three different values of c. For $\beta_N < \beta_{SG}$ inference is asymptotically unfeasible. Two classes of equal size are considered and the entries of J are generated independently according to a Gaussian with mean $J_0\sigma_i\sigma_j$. In the examples, $n = 30\,000$ and averages are over 10 samples.

weighted graphs and, as we explained in Chapter 8, it can be interpreted as a *naïve mean field* (NMF) (hence suboptimal) version of Algorithm 10.2. Also in this case, the comments made in Chapter 8 are valid also in this setting.

Figure 10.7 shows that, while at sufficiently large degree, Algorithm 10.2 and NMF perform essentially equally (as expected), the same cannot be said for a low average degree where the NMF fails severely. Furthermore, also in the weighted case, the Laplacian matrix displays an algorithmic phase transition which is well beyond the one achieved by Algorithm 10.2 and it is hence unsuited for SC in difficult settings. Moreover, note that Algorithm 10.2 achieves a positive overlap as soon as $\beta_N > \beta_{SG}$, *i.e.* it reaches the theoretical detectability threshold and performs competitively with the Bayes optimal solution obtained by *belief propagation* (BP) algorithm. Due to its computational complexity, for $c = 50$, instead of BP we run the TAP algorithm [OS01] that is a convenient approximation of BP to adopt in the presence of a large average degree.

Remark 10.1. *(The "spin glass" Bethe-Hessian) The use of the Bethe-Hessian matrix for SC from sparse pairwise measurements was already proposed in [Saa+16] that studied a similar framework to the one considered in this chapter. In [Saa+16] the temperature $\beta = \beta_{SG}$ is proposed to perform non-trivial clustering as soon as theoretically possible, with a clear reference to the work of [SKZ14] for* community detection *(CD).*

The value β_{SG}, unlike β_N, can be easily estimated from the matrix $\tilde{J}$ solving $c\mathbb{E}[\tanh^2(\beta_{SG}\tilde{J}_{ij}\sigma_i\sigma_j)] = c\mathbb{E}[\tanh^2(\beta_{SG}\tilde{J}_{ij})] = 1$. However, unlike the algorithm of [Saa+16], Algorithm 10.2 represents an optimal relaxation of the Bayes optimal solution and it is expected to yield better performances.

The main difference between the spin glass Bethe-Hessian *and the* Nishimori Bethe-Hessian*, however, is observed when the underlying graph has a heterogeneous degree distribution (see Figure 7 of [DCT21]) and, in the presence of a Erdős-Rényi graph, the performances of both algorithms are observed to be similar, with a slight computational advantage for $H_{\beta_{SG},\tilde{J}}$.*

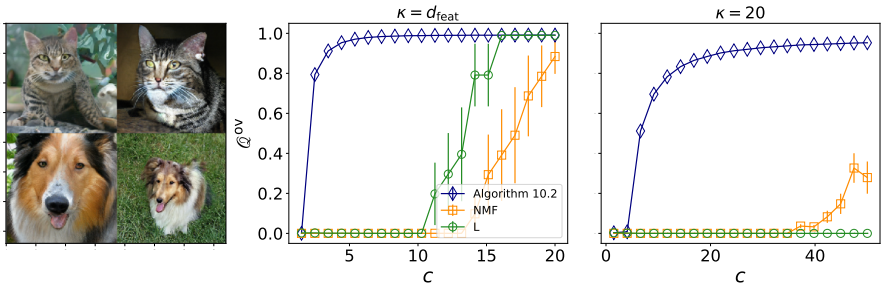


Figure 10.8: **Left plot**: example of random generations of the input GAN images. **Right plots**: overlap classification performance of 40 000 GAN images, as a function of the expected average underlying graph degree c. Here, we consider $K_{ij}(\{z\}) = \frac{1}{d_{\text{feat}}} z_i^T z_j$ and we take either $\kappa = d_{\text{feat}}$: all features of the images are kept, or $\kappa = 20$: on average, only $\sqrt{\kappa / d_{\text{feat}}}$ features ($d_{\text{feat}} = 512$) are used. Averages are taken over 10 realizations.

10.4.2 REAL DATA CLUSTERING

We complete the chapter by a robustness test of our proposed algorithm under a real-world machine learning classification problem. Specifically, we consider a *sparse* (and thus cost-efficient) version of the problem of *correlation clustering* such as met in image classification and show how Algorithm 10.2 can be adopted to accomplish this task with higher performance than with competing spectral methods of the literature.

Given the input vectors stored in the columns of the matrix $Z \in \mathbb{R}^{n \times d_{\text{feat}}}$, we recall from Equation (10.3) that the $\tilde{J}$ is obtained applying a i.i.d. symmetric Bernoulli mask with parameter c/n to the matrix $K = \frac{1}{d_{\text{feat}}} \tilde{Z} \tilde{Z}^T$, where $\tilde{Z} \in \mathbb{R}^{n \times d_{\text{feat}}}$ is obtain discarding random elements of Z, keeping them with probability $\sqrt{\kappa / d_{\text{feat}}}$. We thus practically tested Algorithm 10.2 against the naïve mean field approach which in this setting happens to coincide with the algorithm proposed in [Zar+20] and against the weighted Laplacian matrix.

As a telling modern data classification context, we chose to cluster two classes of high-resolution realistic images randomly produced by *generative adversarial networks* (GANs) [BDS18]; the interest of using GAN images rather than real images lies in that they can be produced "on-the-fly" and in arbitrary numbers. Specifically, we considered $n = 40\,000$ images divided into two groups of equal size, representing collie dogs and tabby cats. A representative example of the input images generated by the GAN is given in Figure 10.8. For each of these images we extracted discriminating features using an off-the-shelf convolutional neural network (VGG) which produces $p = 512$-dimensional feature vectors z_i.[6] We then measured the overlap performance as a function of the average node degree c of the ensuing graph and for different values of κ. The results are reported in Figure 10.8 which strikingly evidences that Algorithm 10.2 can achieve almost perfect recon-

6 The $p = 512$ figure is on the low-hand of typical image vector representations: this number may rise to $4k$ or even to $20k$ when much more than two classes of images are to be classified.

struction already for $c = 5$ when the feature vectors z_i are not sparsified ($\kappa = p$): so, in clearer terms, out of the $40k \times 40k = 1.6 \cdot 10^9$ correlations needed to evaluate the full $K(\{z\})$ matrix, only $\approx 6 \times 40k = 2.4 \cdot 10^5$ is enough to achieve almost optimal performance, thus corresponding to a striking 10^4-fold gain in complexity for a rather marginal performance loss.

10.5 CONCLUSION

The central contribution of this chapter is of a theoretical nature and aims at introducing an explicit relation between the Bethe-Hessian matrix and the Nishimori temperature. Yet, beyond this statistical physics endeavour, the result finds direct applications to Bayesian inference. Specifically, one may anticipate an important impact in more involved applications than those considered in this article, such as in *restricted Boltzmann machines* (RBM) whose goal is to learn a generative model from a set of examples [AHS85]: the Bethe approximation has recently been adopted to study the RBM from a Bayesian perspective [HT16] so that one may envision that the explicit relation between the Bethe free energy and the Nishimori temperature would lead to a better understanding of state-of-the-art algorithms.

On the side of complexity reduction, exploiting high levels of sparsification of data measurements, we showed that our proposed algorithm is capable of accomplishing high quality *unsupervised* classification on very large datasets. This result is all the more fundamental that future machine learning data treatment will call for increasingly larger datasets which cannot be possibly manually labelled and for which unsupervised approaches must be adopted. As a downside though, the generative model we considered for the data affinity (kernel) matrix takes the strong assumption that its entries are drawn from the same probability distribution and only the average (and not the variance, or the distribution itself) embeds information on the node labels. This setting might be too simplistic on generic real data that would require more realistic probability distributions for the generative model of the kernel matrix to be considered.

Possibly most importantly, we worked here under the assumptions that the edges maintained in the sparsified graph are drawn independently at random. When dealing with actual kernel matrices, this cost-efficient measure is quite suboptimal: in [LCM20], a more efficient sparsification procedure is used which maintains the entries of $K(\{z\})$ of largest amplitude. In [LCM20], this comes at the cost of computing all the entries of $K(\{z\})$ but, surely, a more efficient nearest neighbors-type procedure could be implemented as a good performance-complexity compromise [ML09]. Yet, in this setting, although stronger sparsity levels can surely be achieved for the same performance, the key independence property of the entries of $K(\{z\})$ which we exploited here can no longer be assumed, so that one needs to carefully handle the hard problem of dependencies.

CONCLUSION

In this manuscript we investigated *spectral clustering* (SC) techniques adapted (but not necessarily limited) to sparse graphs with applications to *community detection* (CD) on both static and dynamical graphs, as well as correlation clustering of high dimensional vectors. The *file rouge* connecting all our main results revolves around the claim of the existence of isolated informative eigenvalues inside the bulk of the non-backtracking matrix B. Their location is deeply related with the meta-parameters of the generative model of B and the corresponding eigenvectors can be used to obtain a relaxation of the Bayes optimal solution. Furthermore, exploiting its relation with the non-backtracking matrix, we provided cost efficient SC algorithms based on the Bethe-Hessian matrix, extensively tested on real and synthetic datasets.

A profound result of our proposed algorithms relies in their ability to "self-adapt" to the hardness of the clustering problem, bridging existing results of the literature which considered either hard or trivial, either sparse or dense settings. Supported by extensive numerical simulations, we showed that our proposed algorithms often outperform the state-of-the-art "non-adaptive" competing ones.

If on the one hand our work strongly motivates the adoption of well-parametrized Bethe-Hessian matrix for SC, on the other hand it also poses several questions, both on an algorithmic as well as theoretical levels.

LIMITATIONS AND OPEN QUESTIONS

IMPLEMENTATION LIMITATIONS

This manuscript gives a great deal of attention to the problem of designing efficient as well as robust algorithms that can be run on arbitrary input graphs. The empirical experience led us to spot some practical limitations that were not foreseen when we designed our algorithms in the first place. This is particularly the case of Algorithm 7.1 for CD in static graphs.

We recall that Algorithm 7.1 has to compute a sequence of values $\{\zeta_p\}_{p=1,\dots,k}$ satisfying $\lambda_p^{\uparrow}(H_{\zeta_p}) = 0$. To do so, we designed an algorithm that starts from a value $\bar{r} \geq \zeta_k$ to then estimate successively the largest (ζ_k) and arriving to the smallest (ζ_1). The proposed Algorithm 7.3 computes the ζ_p exploiting a relation derived from the Courant-Fischer theorem and does not make any assumption on the function $\lambda_p^{\uparrow}(H_r)$. The practical limitations of this implementation are briefly listed as follows.

1. Suppose that for the input graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, $\zeta_k \approx 1$ and the average degree is sufficiently large. This corresponds to an easy clustering setting (due to $\zeta_k \approx 1$ hypothesis) in which the most commonly adopted SC would likely output a very accurate result. In this case, Algorithm 7.3 has to start from a rather large value $\bar{r}$ (due to the large average degree) and potentially needs several iterations to converge. Consequently, although the performance of Algorithm 7.1 for this easy task is still expected to be very high, its computational time will be significantly higher than the one of competing algorithms.

2. The value $\bar{r} = \sqrt{\lambda_1^{\downarrow|\cdot|}(B)}$ (adopted in Algorithm 7.2) is only *presumed* to be a good approximation of

$$\bar{r} = \arg\max_r \left|\{i \,:\, \lambda_i(H_r) < 0\}\right|.$$

   Algorithm 7.3 requires $\bar{r}$ as an input and the mismatch between its actual and estimated values may potentially lead to discard informative eigenvectors.

3. The number of communities k in a real social graph may grow with its size due to the typical *hierarchical* structure of complex systems [Sim91]. Consequently, on large graphs, very large k may be found. To provide a practical example, the `LiveJournal` social network has $n = 34\,681\,189$ nodes and $k = 287\,512$ communities [YL15]. The computational complexity of Algorithm 7.1 scales as $O(nck^3)$, making it unpractical[1] for a large k like the one of the `LiveJournal` dataset.

We envision that all three problems listed above can be easily solved designing an algorithm that estimates the ζ_p's from the smallest ($\zeta_1 = 1$) to the largest. This strategy obviously solves the problem raised in (1). Also point (2) could be easily addressed, since k can be estimated as the largest p for which ζ_p is defined and has not to be specified before computing the ζ_p's. Concerning point (3), instead, one can envision to adopt a strategy according to which, first the graph is partitioned in k_1 sub-graphs (computing $\zeta_1, \ldots, \zeta_{k_1}$), then on each sub-graph so obtained the process is re-iterated at lower scales until no communities can be detected. For a hierarchical structure, it is reasonable to assume $k_1 \sim \log(k)$, hence solving also problem (3). Unfortunately, we still did not manage to define an algorithm such as Algorithm 7.3 in which the ζ_p's are successively computed from the smallest one, *without* making any assumption on $\lambda_p^{\uparrow}(H_r)$. There lies one of the main improvements of our implementation.

Concerning our proposed algorithms for dynamic CD, on top of the comments detailed in Section 9.5, the main technical issue is certainly related to point (2) of the list above: our result is generally well posed for graphs generated from the *dynamical degree corrected stochastic block model* (DDCSBM),

1 It is worth to remark that, for this particular example, the $O(nk^2)$ complexity of typical SC algorithms may also be prohibitive. Furthermore, making an embedding in $\approx 10^5$ dimensions is unsuited for *k-means*, due to the curse of dimensionality.

but for general graphs, the value of k may be potentially underestimated. The main weakness of Algorithm 10.2 for correlation clustering, instead, resides in point (1): the entries of $H_{\beta,J}$ grow exponentially with β and in the easy clustering regime ($\beta_N \to \infty$) this causes convergence problems. Unlike CD, however, this is not solely a matter of speed of convergence, but also of numerical stability. This could be improved adopting a method like the one proposed in [CD00] to compute the eigenvalues and eigenvectors of matrices with different orders of magnitude.

THE *K-MEANS* STEP

All the results of this manuscript concerned only a specific one of the two steps of SC, that is generating a significant small dimensional node embedding. The whole point of SC, however, is to assign labels and the second step of SC has been systematically performed with the use of *k-means* algorithm. The first question that naturally arises is whether or not this is an optimal choice. We believe that the answer is unlikely to be positive and that more efficient algorithms can be considered in place of *k-means*.

One of the main alternatives to *k-means* is provided by density based clustering algorithms such as [Est+96; Ank+99]. These methods are likely more robust than *k-means* to a asymmetrical embedding such as the one that may appear in Algorithm 7.1 for general graphs. It has to be noted, however, that we experimentally verified that these methods (that typically do not require k as an input) fail to achieve the detectability threshold on synthetic graphs because, close to the transition, there is no region in space in which the density of points is sufficiently low to draw a boundary.

A theoretically more solid alternative (yet less flexible to the implementation on arbitrary graphs) consists in studying the probability distribution of the points in the embedding space and run *expectation maximization* algorithm to perform clustering [DLR77]. Actually, *k-means* consists in solving exactly this problem, assuming that the probability distributions are isotropic Gaussian with the same variance. It has to be noted, however, that to properly address expectation maximization for the problem considered, one needs to have a full theoretical understanding of the eigenvectors of the Bethe-Hessian matrix. To the best of our knowledge, the tools to rigorously address this question in the sparse regime are still lacking.

On a more pragmatical side, Chapters 5, 7 evidenced the striking efficiency of the projection of the embedding vectors on a unitary hypersphere prior to the *k-means* step. This technique is likely to be suboptimal (at least for DCSBM-generated graphs), but we acknowledge that it provides very good results at a moderate computational cost. This evidence, however, needs a thorough theoretical justification, that, in the sparse regime, is still lacking.

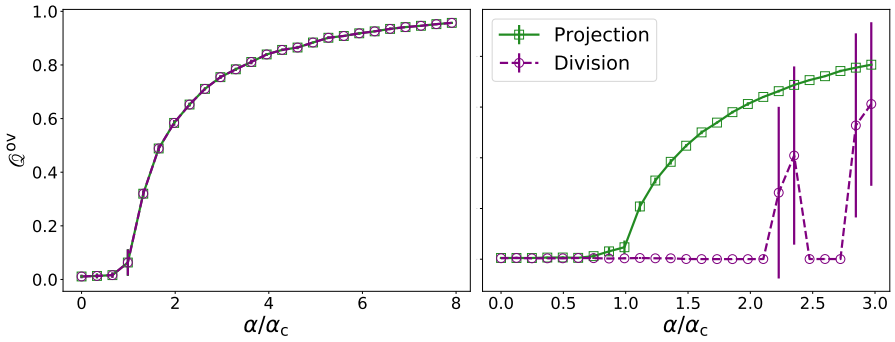


Figure 1: Overlap performance as a function of the detection problem on DCSBM-generated graphs comparing the normalization technique adopted prior to the *k-means* step. The green squares ("projection") are the normalization of the rows of X, while the green dots ("division") the division of the entries of the columns of X by the entries of its first column. **Left**: overlap obtained by Algorithm 4.1 for $M = A$, $n = 10\,000$, $k = 2$, $c = 200$, $\theta_i \sim [\mathcal{U}(3,10)]^4$. **Right**: overlap obtained by Algorithm 4.1 for $M = L_\tau^{\text{sym}}$ and τ equal to the average degree, $n = 25\,000$, $k = 2$, $c = 5$, $\theta_i \sim [\mathcal{U}(3,10)]^4$. Both simulations are averaged over 10 realizations.

To motivate this assertion, we recall that the projection step is justified when the entries of the embedding matrix $X \in \mathbb{R}^{n\times k}$ can be written as

$$X_{i,\bullet} = [f_i g_{\ell_i}^{(1)}, f_i g_{\ell_i}^{(2)}, \dots, f_i g_{\ell_i}^{(k)}].$$

In practice, the entries of $X_{i,\bullet}$ are decomposed in two contributions: one that remains constant along the row but that is different for each node, f_i; one that takes a different value for each column of X, but depends on i only through its label ℓ_i and is therefore capable to group together nodes into communities. Normalizing $\|X_{i,\bullet}\| = 1$, we see that the i-dependent component of $X_{i,\bullet}$ is removed, producing a high quality embedding which depends solely on the node labels. An alternative strategy to preprocess X however exists and it consists in defining $\tilde{X}$ as follows

$$\tilde{X}_{\bullet,i} = \frac{X_{\bullet,i}}{X_{\bullet,1}}. \tag{1}$$

Although these two normalization procedures may seem very similar, in practice they perform very differently. In particular, the left plot of Figure 1 shows the performance of clustering on the leading eigenvectors A on dense DCSBM graphs and for the two normalization strategies. Note that the normalization obtained from Equation (1) precisely corresponds to the *SCORE* algorithm of [Jin+15]. Figure 1 shows that the two methods perform equally well, thus fully justifying the projection step in this case.

On the right, the same plot is proposed in the sparse regime for $L_\tau^{\text{sym}} = D_\tau^{-1/2} A D_\tau^{-1/2}$ for τ equal the average degree. In this case, adopting the projection on the hypersphere as a normalization corresponds to the algorithm of [QR13], which we extensively commented in Chapter 6. In this case, the two plots are in evident disagreement, showing that the high performance obtained from the projection step goes beyond its theoretical justification.

Also in this case, to formally address this question, one would need to have clear results on the eigenvectors of the considered matrix to then study the convergence of *k-means* on the obtained embedding.

To conclude, our results address the problem of designing a proper node embedding with a SC algorithm. Nonetheless, SC requires also to be capable of extracting the information from the embedding and considering the problem as a whole certainly deserves further investigations.

THEORETICAL CONSIDERATIONS

The work presented in this manuscript raises some important questions also from a theoretical standpoint. Perhaps the most relevant result concerns an explicit relation between the spectral properties of the Bethe-Hessian matrix and the Nishimori temperature as detailed in Chapter 10. It has to be noted that, although the mapping between the Ising model and Bayesian inference was fully rigorous only in Chapter 10, the results of both dynamic and static CD are closely related.

From a statistical physics perspective, a question that certainly deserves further exploration concerns the deep meaning and implications of the Bethe-Nishimori relation. From our conjecture, it appears that the paramagnetic point $\hat{\boldsymbol{m}} = \mathbf{0}_n$ is not a minimum of the Bethe free energy for $\beta_{\mathrm{F}} < \beta < \beta_{\mathrm{N}}$, but it is a local minimum for $\beta > \beta_{\mathrm{N}}$, with the Nishimori temperature denoting the transition between these two conditions. At low temperature, $\hat{\boldsymbol{m}} = \mathbf{0}_n$ corresponds to a *local* minimum of the free energy hence to a configuration that is metastable. This result should be related to existing experimental and numerical observations of the *random bond Ising model* (RBIM).

On top of this, the Nishimori-Bethe relation is formulated only for *Erdős-Rényi* (ER) random graphs and for Ising variables. The properties of the Nishimori temperature can however be extended to a larger class of models than the RBIM [Geo+85] and to graphs not generated from the ER model. The natural line of exploration in this direction is hence to study the relation between the Bethe free energy and the Nishimori temperature for different classes of graphs and Hamiltonians.

Moreover, beyond the interest in statistical physics of our results, we also unveiled some previously unseen behavior of the non-backtracking matrix eigenvalues, predicting the existence of isolated eigenvalues *inside* its bulk. From an algebraic standpoint, the emergence of these eigenvalues is well understood in quasi-regular graphs [CZ20], but we believe that this explanation does not capture the depth and complexity of the problem. Further mathematical tools would then be needed to formally address this question and study these eigenvalues and their related eigenvectors.

OUTLOOK AND PERSPECTIVES

Let us now look ahead, proposing some further lines of explorations and applications of our results.

Concerning the problem of CD we detailed extensively in Chapters 7, 8 that our proposed Algorithm 7.1 can efficiently be implemented on arbitrary graphs, but its backbone is deeply motivated by the DCSBM which assumes that the edges of the graph are generated independently at random. In typical realizations of social graphs, however, the entries of the adjacency matrix are highly correlated making indeed questionable the DCSBM assumption in practice. In order to keep into account these additional dependences induced by the presence of short cycles, in [KCN21] the authors recently proposed an improved version of *belief propagation* (BP) algorithm, adapted to operate in the presence of short cycles, further defining the related non-backtracking matrix. Other "higher order approximations" exist to properly deal with the non tree-like structure of a graph. To accomplish this task, a valid alternative to the Bethe is the *cluster variational method* [Pelo5].

The capability of properly treating dependencies is not constrained to CD. In fact, in the context of correlation clustering, we discussed that the i.i.d. sparsification strategy is suboptimal in terms of performance, indeed because it makes the entries of the kernel matrix asymptotically independent. More efficient sampling techniques may be designed achieving a good level of sparsity, but keeping the correlations among the entries of the matrix. The adoption of proper variational approximations to deal at once with dependencies and sparsity is certainly key to design efficient algorithms.

Most of our results revolve around the very popular (but somewhat simplistic) problem of CD. In the chapters of Part III we however showed how the intuitions and methods developed for CD can be easily extended to more involved settings. This unveils on the one hand the power of the method and on the other hand the depth of Claim 5.1, which is formulated for the DCSBM, but that conveys an intuition that goes well beyond its assumptions.

Most crucially, based on this observation and the discussions of Chapter 8, we envision that the "Bethe-Hessian" paradigm should be systematically considered as an alternative to the classical Laplacian and "weighted adjacency" matrices that are commonly adopted for several machine learning applications. In fact, we showed that the most popular techniques for SC can be interpreted as limiting cases of our proposed algorithms. The huge advantage of our algorithms is that they constitute a bridge between these extremes and consequently perform better at practical SC problems.

With this in mind, we close this manuscript with a question that invites the practitioner to reflect any time the eigenvectors of a weighted adjacency or Laplacian matrix are exploited in one of the countless applications, whether it is possible to define a more appropriate graph matrix representation introducing a proper temperature parameter or a better variational approximation to obtain improved performances.

PUBLICATIONS

1. **LD**, Romain Couillet: *Community detection in sparse realistic graphs: Improving the Bethe-Hessian* in ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (2019)

2. **LD**, Romain Couillet, Nicolas Tremblay: *Classification spectrale par la laplacienne déformée dans des graphes réalistes* in XXVIIème colloque GRETSI (2019)

3. **LD**, Romain Couillet, Nicolas Tremblay: *Revisiting the Bethe-Hessian: Improved Community Detection in Sparse Heterogeneous Graphs* in NeurIPS Advances in Neural Information Processing Systems 32 (2019)

4. **LD**, Romain Couillet, Nicolas Tremblay: *Optimal Laplacian regularization for sparse spectral community detection* in ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (2020)

5. **LD**, Romain Couillet, Nicolas Tremblay: *A unified framework for spectral clustering in sparse graphs* accepted to Journal of machine learning research

6. **LD**, Romain Couillet, Nicolas Tremblay: *Community detection in sparse time-evolving graphs with a dynamical Bethe-Hessian* in NeurIPS Advances in Neural Information Processing Systems 33 (2020)

7. **LD**, Romain Couillet, Nicolas Tremblay: *Nishimori meets Bethe: a spectral method for node classification in sparse weighted graphs* in Journal of Statistical Mechanics: Theory and Experiment

BIBLIOGRAPHY

[Abb17] Emmanuel Abbe. "Community detection and stochastic block models: recent developments." In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 6446–6531.

[AS15] Emmanuel Abbe and Colin Sandon. "Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic bp, and the information-computation gap." In: *arXiv preprint arXiv:1512.09080* (2015).

[AHS85] David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. "A learning algorithm for Boltzmann machines." In: *Cognitive science* 9.1 (1985), pp. 147–169.

[AG05] Lada A Adamic and Natalie Glance. "The political blogosphere and the 2004 US election: divided they blog." In: *Proceedings of the 3rd international workshop on Link discovery*. 2005, pp. 36–43.

[AB02] Réka Albert and Albert-László Barabási. "Statistical mechanics of complex networks." In: *Reviews of modern physics* 74.1 (2002), p. 47.

[AC17] Hafiz Tiomoko Ali and Romain Couillet. "Improved spectral community detection in large heterogeneous networks." In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 8344–8392.

[AA03] Eric Alm and Adam P Arkin. "Biological networks." In: *Current opinion in structural biology* 13.2 (2003), pp. 193–202.

[Ama+00] Luıs A Nunes Amaral, Antonio Scala, Marc Barthelemy, and H Eugene Stanley. "Classes of small-world networks." In: *Proceedings of the national academy of sciences* 97.21 (2000), pp. 11149–11152.

[Ami+13] Arash A Amini, Aiyou Chen, Peter J Bickel, Elizaveta Levina, et al. "Pseudo-likelihood methods for community detection in large sparse networks." In: *Annals of Statistics* 41.4 (2013), pp. 2097–2122.

[Ank+99] Mihael Ankerst, Markus M Breunig, Hans-Peter Kriegel, and Jörg Sander. "OPTICS: Ordering points to identify the clustering structure." In: *ACM Sigmod record* 28.2 (1999), pp. 49–60.

[AFG08] Alex Arenas, Alberto Fernandez, and Sergio Gomez. "Analysis of the structure of complex networks at different resolution levels." In: *New journal of physics* 10.5 (2008), p. 053039.

[AV06] David Arthur and Sergei Vassilvitskii. *k-means++: The advantages of careful seeding*. Tech. rep. Stanford, 2006.

[Ban+15] Natalie Clare Banks, Dean Ronald Paini, Kirsty Louise Bayliss, and Michael Hodda. "The role of global trade and transport network topology in the human-mediated dispersal of alien species." In: *Ecology letters* 18.2 (2015), pp. 188–199.

[BBC04] Nikhil Bansal, Avrim Blum, and Shuchi Chawla. "Correlation clustering." In: *Machine learning* 56.1 (2004), pp. 89–113.

[Bar03] Albert-László Barabási. *Linked: The new science of networks*. 2003.

[Bar13] Albert-László Barabási. "Network science." In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 371.1987 (2013), p. 20120375.

[BA99] Albert-László Barabási and Réka Albert. "Emergence of scaling in random networks." In: *science* 286.5439 (1999), pp. 509–512.

[Bar+04] Alain Barrat, Marc Barthelemy, Romualdo Pastor-Satorras, and Alessandro Vespignani. "The architecture of complex weighted networks." In: *Proceedings of the national academy of sciences* 101.11 (2004), pp. 3747–3752.

[BC13] Alain Barrat and Ciro Cattuto. "Temporal networks of face-to-face human interactions." In: *Temporal Networks*. Springer, 2013, pp. 191–216.

[Bar+13] Alain Barrat, Ciro Cattuto, Vittoria Colizza, Francesco Gesualdo, Lorenzo Isella, Elisabetta Pandolfi, J-F Pinton, Lucilla Ravà, Caterina Rizzo, Mariateresa Romano, et al. "Empirical temporal networks of face-to-face human interactions." In: *The European Physical Journal Special Topics* 222.6 (2013), pp. 1295–1309.

[Bar11] Marc Barthélemy. "Spatial networks." In: *Physics Reports* 499.1-3 (2011), pp. 1–101.

[BF60] Friedrich L Bauer and Charles T Fike. "Norms and exclusion theorems." In: *Numerische Mathematik* 2.1 (1960), pp. 137–141.

[Bel15] Richard E Bellman. *Adaptive control processes: a guided tour*. Princeton university press, 2015.

[BGBK+19] Florent Benaych-Georges, Charles Bordenave, Antti Knowles, et al. "Largest eigenvalues of sparse inhomogeneous Erdős–Rényi graphs." In: *Annals of Probability* 47.3 (2019), pp. 1653–1676.

[BGBK+20] Florent Benaych-Georges, Charles Bordenave, Antti Knowles, et al. "Spectral radii of sparse random matrices." In: *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*. Vol. 56. 3. Institut Henri Poincaré. 2020, pp. 2141–2161.

[Bet35] Hans A Bethe. "Statistical theory of superlattices." In: *Proceedings of the Royal Society of London. Series A-Mathematical and Physical Sciences* 150.871 (1935), pp. 552–575.

[Bha13] Rajendra Bhatia. *Matrix analysis*. Vol. 169. Springer Science & Business Media, 2013.

[BY86] Kurt Binder and A Peter Young. "Spin glasses: Experimental facts, theoretical concepts, and open questions." In: *Reviews of Modern physics* 58.4 (1986), p. 801.

[Bis06] Christopher M Bishop. *Pattern recognition and machine learning*. springer, 2006.

[Blo+08] Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. "Fast unfolding of communities in large networks." In: *Journal of statistical mechanics: theory and experiment* 2008.10 (2008), P10008.

[BB01] Béla Bollobás and Bollobás Béla. *Random graphs*. 73. Cambridge university press, 2001.

[BJR07] Béla Bollobás, Svante Janson, and Oliver Riordan. "The phase transition in inhomogeneous random graphs." In: *Random Structures & Algorithms* 31.1 (2007), pp. 3–122.

[BLM15] Charles Bordenave, Marc Lelarge, and Laurent Massoulié. "Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs." In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE. 2015, pp. 1347–1357.

[BH11] Stephen P Borgatti and Daniel S Halgin. "On network theory." In: *Organization science* 22.5 (2011), pp. 1168–1181.

[Bor+13] Pierre Borgnat, Céline Robardet, Patrice Abry, Patrick Flandrin, Jean-Baptiste Rouquier, and Nicolas Tremblay. "A dynamical network view of lyon's vélo'v shared bicycle system." In: *Dynamics On and Of Complex Networks, Volume 2*. Springer, 2013, pp. 267–284.

[BP86] Ronald L Breiger and Philippa E Pattison. "Cumulated social roles: The duality of persons and their algebras." In: *Social networks* 8.3 (1986), pp. 215–256.

[BDS18] Andrew Brock, Jeff Donahue, and Karen Simonyan. "Large scale GAN training for high fidelity natural image synthesis." In: *arXiv preprint arXiv:1809.11096* (2018).

[Bru76a] Stephen G Brush. "Statistical mechanics and the philosophy of science: some historical notes." In: *PSA: Proceedings of the Biennial Meeting of the Philosophy of Science Association*. Vol. 1976. 2. Philosophy of Science Association. 1976, pp. 551–584.

[Bru76b] Stephen G Brush. *The kind of motion we call heat*. Vol. 2. North-Holland Amsterdam, 1976.

[CZC18] Hongyun Cai, Vincent W Zheng, and Kevin Chen-Chuan Chang. "A comprehensive survey of graph embedding: Problems, techniques, and applications." In: *IEEE Transactions on Knowledge and Data Engineering* 30.9 (2018), pp. 1616–1637.

[Car+19] Giuseppe Carleo, Ignacio Cirac, Kyle Cranmer, Laurent Daudet, Maria Schuld, Naftali Tishby, Leslie Vogt-Maranto, and Lenka Zdeborová. "Machine learning and the physical sciences." In: *Reviews of Modern Physics* 91.4 (2019), p. 045002.

[CFL09] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. "Statistical physics of social dynamics." In: *Reviews of modern physics* 81.2 (2009), p. 591.

[Cat+10] Ciro Cattuto, Wouter Van den Broeck, Alain Barrat, Vittoria Colizza, Jean-François Pinton, and Alessandro Vespignani. "Dynamics of person-to-person interactions from distributed RFID sensor networks." In: *PloS one* 5.7 (2010), e11596.

[CR19] Remy Cazabet and Giulio Rossetti. "Challenges in community discovery on temporal networks." In: *Temporal Network Theory*. Springer, 2019, pp. 181–197.

[CD00] Tzu-Yi Chen and James W Demmel. "Balancing sparse matrices for computing eigenvalues." In: *Linear algebra and its applications* 309.1-3 (2000), pp. 261–287.

[Chi+07] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng. "Evolutionary spectral clustering by incorporating temporal smoothness." In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2007, pp. 153–162.

[Chi+09] Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L Tseng. "On evolutionary spectral clustering." In: *ACM Transactions on Knowledge Discovery from Data (TKDD)* 3.4 (2009), pp. 1–30.

[CG97] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. 92. American Mathematical Soc., 1997.

[CLV04] Fan Chung, Linyuan Lu, and Van Vu. "The spectra of random graphs with given expected degrees." In: *Internet Mathematics* 1.3 (2004), pp. 257–275.

[Cla34] Émile Clapeyron. "Mémoire sur la puissance motrice de la chaleur." In: *Journal de l'École polytechnique* 14 (1834), pp. 153–190.

[CNM04] Aaron Clauset, Mark EJ Newman, and Cristopher Moore. "Finding community structure in very large networks." In: *Physical review E* 70.6 (2004), p. 066111.

[CO+18] Amin Coja-Oghlan, Florent Krzakala, Will Perkins, and Lenka Zdeborová. "Information-theoretic thresholds from the cavity method." In: *Advances in Mathematics* 333 (2018), pp. 694–795.

[CZ20] Simon Coste and Yizhe Zhu. "Eigenvalues of the non-backtracking operator detached from the bulk." In: *Random Matrices: Theory and Applications* (2020), p. 2150028.

[CBG+16] Romain Couillet, Florent Benaych-Georges, et al. "Kernel spectral clustering of large dimensional data." In: *Electronic Journal of Statistics* 10.1 (2016), pp. 1393–1454.

[CCB21] Romain Couillet, Florent Chatelain, and Nicolas Le Bihan. "Two-way kernel matrix puncturing: towards resource-efficient PCA and spectral clustering." In: *International Conference on Machine Learning* (2021).

[CD11] Romain Couillet and Merouane Debbah. *Random matrix methods for wireless communications*. Cambridge University Press, 2011.

[Cov99] Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

[CG11] Bernard Dennis Cullity and Chad D Graham. *Introduction to magnetic materials*. John Wiley & Sons, 2011.

[CRS07] Dragoš Cvetković, Peter Rowlinson, and Slobodan K Simić. "Signless Laplacians of finite graphs." In: *Linear Algebra and its applications* 423.1 (2007), pp. 155–171.

[DCT19] Lorenzo Dall'Amico, Romain Couillet, and Nicolas Tremblay. "Revisiting the Bethe-Hessian: Improved Community Detection in Sparse Heterogeneous Graphs." In: *Thirty-third Conference on Neural Information Processing Systems*. 2019.

[DCT20a] Lorenzo Dall'Amico, Romain Couillet, and Nicolas Tremblay. "A unified framework for spectral clustering in sparse graphs." In: *arXiv preprint arXiv:2003.09198* (2020).

[DCT20b] Lorenzo Dall'Amico, Romain Couillet, and Nicolas Tremblay. "Community detection in sparse time-evolving graphs with a dynamical Bethe-Hessian." In: *Advances in Neural Information Processing Systems* 33 (2020).

[DCT21] Lorenzo Dall'Amico, Romain Couillet, and Nicolas Tremblay. "Nishimori meets Bethe: a spectral method for node classification in sparse weighted graphs." In: *Journal of Statistical Mechanics: Theory and Experiment* 2021.9 (2021), p. 093405.

[DC19] Lorenzo Dall'Amico and Romain Couillet. "Community detection in sparse realistic graphs: Improving the bethe hessian." In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 2942–2946.

[DCT20c] Lorenzo Dall'Amico, Romain Couillet, and Nicolas Tremblay. "Optimal Laplacian regularization for sparse spectral community detection." In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 3237–3241.

[Dec+11] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. "Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications." In: *Physical Review E* 84.6 (2011), p. 066106.

[DM+10a] Amir Dembo, Andrea Montanari, et al. "Gibbs measures and phase transitions on sparse random graphs." In: *Brazilian Journal of Probability and Statistics* 24.2 (2010), pp. 137–211.

[DM+10b] Amir Dembo, Andrea Montanari, et al. "Ising models on locally tree-like graphs." In: *The Annals of Applied Probability* 20.2 (2010), pp. 565–592.

[DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. "Maximum likelihood from incomplete data via the EM algorithm." In: *Journal of the Royal Statistical Society: Series B (Methodological)* 39.1 (1977), pp. 1–22.

[DGM08] Sergey N Dorogovtsev, Alexander V Goltsev, and José FF Mendes. "Critical phenomena in complex networks." In: *Reviews of Modern Physics* 80.4 (2008), p. 1275.

[DM02] Sergey N Dorogovtsev and Jose FF Mendes. "Evolution of networks." In: *Advances in physics* 51.4 (2002), pp. 1079–1187.

[EP06] Nathan Eagle and Alex Sandy Pentland. "Reality mining: sensing complex social systems." In: *Personal and ubiquitous computing* 10.4 (2006), pp. 255–268.

[EA75] Samuel Frederick Edwards and Phil W Anderson. "Theory of spin glasses." In: *Journal of Physics F: Metal Physics* 5.5 (1975), p. 965.

[ER60] Paul Erdős and Alfréd Rényi. "On the evolution of random graphs." In: *Publ. Math. Inst. Hung. Acad. Sci* 5.1 (1960), pp. 17–60.

[Est+96] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *Kdd*. Vol. 96. 34. 1996, pp. 226–231.

[Fie73] Miroslav Fiedler. "Algebraic connectivity of graphs." In: *Czechoslovak mathematical journal* 23.2 (1973), pp. 298–305.

[For10] Santo Fortunato. "Community detection in graphs." In: *Physics reports* 486.3-5 (2010), pp. 75–174.

[FB07] Santo Fortunato and Marc Barthelemy. "Resolution limit in community detection." In: *Proceedings of the national academy of sciences* 104.1 (2007), pp. 36–41.

[FH16] Santo Fortunato and Darko Hric. "Community detection in networks: A user guide." In: *Physics reports* 659 (2016), pp. 1–44.

[GBC14] Valerio Gemmetto, Alain Barrat, and Ciro Cattuto. "Mitigation of infectious disease at school: targeted class closure vs school closure." In: *BMC infectious diseases* 14.1 (2014), pp. 1–10.

[Geo+85] Antoine Georges, David Hansel, Pierre Le Doussal, and J-P Bouchaud. "Exact properties of spin glasses. II. Nishimori's line: new results and physical implications." In: *Journal de Physique* 46.11 (1985), pp. 1827–1836.

[Gha+16] Amir Ghasemian, Pan Zhang, Aaron Clauset, Cristopher Moore, and Leto Peel. "Detectability thresholds and optimal algorithms for community structure in dynamic networks." In: *Physical Review X* 6.3 (2016), p. 031005.

[Gho06] Ali Ghodsi. "Dimensionality reduction a short tutorial." In: *Department of Statistics and Actuarial Science, Univ. of Waterloo, Ontario, Canada* 37.38 (2006), p. 2006.

[Gib14] J Willard Gibbs. *Elementary principles in statistical mechanics*. Courier Corporation, 2014.

[GN02] Michelle Girvan and Mark EJ Newman. "Community structure in social and biological networks." In: *Proceedings of the national academy of sciences* 99.12 (2002), pp. 7821–7826.

[GDMC10] Benjamin H Good, Yves-Alexandre De Montjoye, and Aaron Clauset. "Performance of modularity maximization in practical contexts." In: *Physical Review E* 81.4 (2010), p. 046106.

[GF18] Palash Goyal and Emilio Ferrara. "Graph embedding techniques, applications, and performance: A survey." In: *Knowledge-Based Systems* 151 (2018), pp. 78–94.

[GLO20] Anne Greenbaum, Ren-cang Li, and Michael L Overton. "First-order perturbation theory for eigenvalues and eigenvectors." In: *SIAM review* 62.2 (2020), pp. 463–482.

[GRL01] Ilya A Gruzberg, N Read, and Andreas WW Ludwig. "Random-bond Ising model in two dimensions: The Nishimori line and supersymmetry." In: *Physical Review B* 63.10 (2001), p. 104422.

[Gua90] John Guare. *Six degrees of separation: A play*. Vintage, 1990.

[GSPA04] Roger Guimera, Marta Sales-Pardo, and Luís A Nunes Amaral. "Modularity from fluctuations in random graphs and complex networks." In: *Physical Review E* 70.2 (2004), p. 025101.

[GLM17a] Lennart Gulikers, Marc Lelarge, and Laurent Massouli. "Non-Backtracking Spectrum of Degree-Corrected Stochastic Block Models." In: *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*. Vol. 67. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 2017, p. 44.

[GLM15] Lennart Gulikers, Marc Lelarge, and Laurent Massoulié. "An impossibility result for reconstruction in a degree-corrected planted-partition model." In: *arXiv preprint arXiv:1511.00546* (2015).

[GLM16] Lennart Gulikers, Marc Lelarge, and Laurent Massoulié. "Non-backtracking spectrum of degree-corrected stochastic block models." In: *arXiv preprint arXiv:1609.02487* (2016).

[GLM17b] Lennart Gulikers, Marc Lelarge, and Laurent Massoulié. "A spectral method for community detection in moderately sparse degree-corrected stochastic block models." In: *Advances in Applied Probability* (2017), pp. 686–721.

[HSS08] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. "Exploring Network Structure, Dynamics, and Function using NetworkX." In: *Proceedings of the 7th Python in Science Conference*. Ed. by Gaël Varoquaux, Travis Vaught, and Jarrod Millman. Pasadena, CA USA, 2008, pp. 11 –15.

[HK92] Lars Hagen and Andrew B Kahng. "New spectral methods for ratio cut partitioning and clustering." In: *IEEE transactions on computer-aided design of integrated circuits and systems* 11.9 (1992), pp. 1074–1085.

[Ham+04] Jihun Ham, Daniel D Lee, Sebastian Mika, and Bernhard Schölkopf. "A kernel view of the dimensionality reduction of manifolds." In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 47.

[Has89] Ki-ichiro Hashimoto. "Zeta functions of finite graphs and representations of p-adic groups." In: *Automorphic forms and geometry of arithmetic varieties*. Elsevier, 1989, pp. 211–280.

[Has70] W Keith Hastings. "Monte Carlo sampling methods using Markov chains and their applications." In: (1970).

[Hes+03] Tom Heskes et al. "Stable fixed points of loopy belief propagation are minima of the Bethe free energy." In: *Advances in neural information processing systems* 15 (2003), pp. 359–366.

[HLL83] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. "Stochastic blockmodels: First steps." In: *Social networks* 5.2 (1983), pp. 109–137.

[HL71] Paul W Holland and Samuel Leinhardt. "Transitivity in structural models of small groups." In: *Comparative group studies* 2.2 (1971), pp. 107–124.

[Hol15] Petter Holme. "Modern temporal network theory: a colloquium." In: *The European Physical Journal B* 88.9 (2015), pp. 1–30.

[HS12] Petter Holme and Jari Saramäki. "Temporal networks." In: *Physics reports* 519.3 (2012), pp. 97–125.

[HJ12] Roger A Horn and Charles R Johnson. *Matrix analysis*. Cambridge university press, 2012.

[HST06] Matthew D Horton, HM Stark, and Audrey A Terras. "What are zeta functions of graphs and what are they good for?" In: *Contemporary Mathematics* 415 (2006), pp. 173–190.

[HT16] Haiping Huang and Taro Toyoizumi. "Unsupervised feature learning from finite data by message passing: discontinuous versus continuous phase transition." In: *Physical Review E* 94.6 (2016), p. 062310.

[Hul20] Clark L Hull. "Quantitative aspects of evolution of concepts: An experimental study." In: *Psychological monographs* 28.1 (1920), p. i.

[Iba99] Yukito Iba. "The Nishimori line and Bayesian statistics." In: *Journal of Physics A: Mathematical and General* 32.21 (1999), p. 3875.

[Isi25] Ernst Ising. "Contribution to the theory of ferromagnetism." In: *Z. Phys* 31.1 (1925), pp. 253–258.

[JM+04] Svante Janson, Elchanan Mossel, et al. "Robust reconstruction on trees is determined by the second eigenvalue." In: *Annals of probability* 32.3B (2004), pp. 2630–2649.

[Jay+99] Laurent O Jay, Hanchul Kim, Yousef Saad, and James R Chelikowsky. "Electronic structure calculations for plane-wave codes without diagonalization." In: *Computer physics communications* 118.1 (1999), pp. 21–30.

[Jin+15] Jiashun Jin et al. "Fast community detection by score." In: *Annals of Statistics* 43.1 (2015), pp. 57–89.

[JH85] Charles R Johnson and Roger A Horn. *Matrix analysis*. Cambridge university press Cambridge, 1985.

[JY13] Antony Joseph and Bin Yu. "Impact of regularization on spectral clustering." In: *arXiv preprint arXiv:1312.1733* (2013).

[JY+16] Antony Joseph, Bin Yu, et al. "Impact of regularization on spectral clustering." In: *Annals of Statistics* 44.4 (2016), pp. 1765–1791.

[KN11] Brian Karrer and Mark EJ Newman. "Stochastic blockmodels and community structure in networks." In: *Physical review E* 83.1 (2011), p. 016107.

[KNZ14] Brian Karrer, Mark EJ Newman, and Lenka Zdeborová. "Percolation on sparse networks." In: *Physical review letters* 113.20 (2014), p. 208702.

[KR09] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*. Vol. 344. John Wiley & Sons, 2009.

[KV20] Nicolas Keriven and Samuel Vaiter. "Sparse and smooth: improved guarantees for spectral clustering in the dynamic stochastic block model." In: *arXiv preprint arXiv:2002.02892* (2020).

[KCN21] Alec Kirkley, George T Cantwell, and MEJ Newman. "Belief propagation for networks with loops." In: *Science Advances* 7.17 (2021), eabf1211.

[Kiv+14] Mikko Kivelä, Alex Arenas, Marc Barthelemy, James P Gleeson, Yamir Moreno, and Mason A Porter. "Multilayer networks." In: *Journal of complex networks* 2.3 (2014), pp. 203–271.

[KPW13] Martin Kochmański, Tadeusz Paszkiewicz, and Sławomir Wolski. "Curie–Weiss magnet—a simple model of phase transition." In: *European Journal of Physics* 34.6 (2013), p. 1555.

[KKZ09] Hans-Peter Kriegel, Peer Kröger, and Arthur Zimek. "Clustering high-dimensional data: A survey on subspace clustering, pattern-based clustering, and correlation clustering." In: *Acm transactions on knowledge discovery from data (tkdd)* 3.1 (2009), pp. 1–58.

[Krz+13] Florent Krzakala, Cristopher Moore, Elchanan Mossel, Joe Neeman, Allan Sly, Lenka Zdeborová, and Pan Zhang. "Spectral redemption in clustering sparse networks." In: *Proceedings of the National Academy of Sciences* 110.52 (2013), pp. 20935–20940.

[KLB09] Jérôme Kunegis, Andreas Lommatzsch, and Christian Bauckhage. "The slashdot zoo: mining a social network with negative edges." In: *Proceedings of the 18th international conference on World wide web*. 2009, pp. 741–750.

[Kun+10] Jérôme Kunegis, Stephan Schmidt, Andreas Lommatzsch, Jürgen Lerner, Ernesto W De Luca, and Sahin Albayrak. "Spectral analysis of signed graphs for clustering, prediction and visualization." In: *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM. 2010, pp. 559–570.

[Lan+16] Rocco Langone, Raghvendra Mall, Carlos Alzate, and Johan AK Suykens. "Kernel spectral clustering and applications." In: *Unsupervised Learning Algorithms*. Springer, 2016, pp. 135–161.

[LM+54] Paul F Lazarsfeld, Robert K Merton, et al. "Friendship as a social process: A substantive and methodological analysis." In: *Freedom and control in modern society* 18.1 (1954), pp. 18–66.

[LL15] Can M Le and Elizaveta Levina. "Estimating the number of communities in networks by spectral methods." In: *arXiv preprint arXiv:1507.00827* (2015).

[LLV15] Can M Le, Elizaveta Levina, and Roman Vershynin. "Sparse random graphs: regularization and concentration of the laplacian." In: *arXiv preprint arXiv:1502.03049* (2015).

[LLV18] Can M Le, Elizaveta Levina, and Roman Vershynin. *Concentration of random graphs and application to community detection*. World Scientific, 2018.

[Leb+10] Sophie Lebre, Jennifer Becq, Frederic Devaux, Michael PH Stumpf, and Gaelle Lelandais. "Statistical inference of the time-varying structure of gene-regulation networks." In: *BMC systems biology* 4.1 (2010), pp. 1–16.

[LR+15] Jing Lei, Alessandro Rinaldo, et al. "Consistency of spectral clustering in stochastic block models." In: *Annals of Statistics* 43.1 (2015), pp. 215–237.

[Leo+02] M Leone, A Vázquez, A Vespignani, and Riccardo Zecchina. "Ferromagnetic ordering in graphs with arbitrary degree distribution." In: *The European Physical Journal B-Condensed Matter and Complex Systems* 28.2 (2002), pp. 191–197.

[LK14] Jure Leskovec and Andrej Krevl. *SNAP Datasets: Stanford large network dataset collection*. 2014.

[LCM20] Zhenyu Liao, Romain Couillet, and Michael W Mahoney. "Sparse quantized spectral clustering." In: *arXiv preprint arXiv:2010.01376* (2020).

[Liu+18] Fuchen Liu, David Choi, Lu Xie, and Kathryn Roeder. "Global spectral clustering in dynamic networks." In: *Proceedings of the National Academy of Sciences* 115.5 (2018), pp. 927–932.

[Llo82] Stuart Lloyd. "Least squares quantization in PCM." In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.

[Lus03] David Lusseau. "The emergent properties of a dolphin social network." In: *Proceedings of the Royal Society of London. Series B: Biological Sciences* 270.suppl_2 (2003), S186–S188.

[Lus+03] David Lusseau, Karsten Schneider, Oliver J Boisseau, Patti Haase, Elisabeth Slooten, and Steve M Dawson. "The bottlenose dolphin community of Doubtful Sound features a large proportion of long-lasting associations." In: *Behavioral Ecology and Sociobiology* 54.4 (2003), pp. 396–405.

[Mac+67] James MacQueen et al. "Some methods for classification and analysis of multivariate observations." In: *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*. Vol. 1. 14. Oakland, CA, USA. 1967, pp. 281–297.

[Mas14] Laurent Massoulié. "Community detection thresholds and the weak Ramanujan property." In: *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*. 2014, pp. 694–703.

[Max67] James Clerk Maxwell. "IV. On the dynamical theory of gases." In: *Philosophical transactions of the Royal Society of London* 157 (1867), pp. 49–88.

[MP43] Warren S McCulloch and Walter Pitts. "A logical calculus of the ideas immanent in nervous activity." In: *The bulletin of mathematical biophysics* 5.4 (1943), pp. 115–133.

[MSLC01] Miller McPherson, Lynn Smith-Lovin, and James M Cook. "Birds of a feather: Homophily in social networks." In: *Annual review of sociology* 27.1 (2001), pp. 415–444.

[MM09] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.

[MPV87] Marc Mézard, Giorgio Parisi, and Miguel Angel Virasoro. *Spin glass theory and beyond: An Introduction to the Replica Method and Its Applications*. Vol. 9. World Scientific Publishing Company, 1987.

[Mil67] Stanley Milgram. "The small world problem." In: *Psychology today* 2.1 (1967), pp. 60–67.

[MRD21] Alexander Modell and Patrick Rubin-Delanchy. "Spectral clustering under degree heterogeneity: a case for the random walk Laplacian." In: *arXiv preprint arXiv:2105.00987* (2021).

[MRT18] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

[Moo17] Cristopher Moore. "The computer science and physics of community detection: Landscapes, phase transitions, and hardness." In: *arXiv preprint arXiv:1702.00467* (2017).

[Mos04] Elchanan Mossel. "Survey: Information flow on trees." In: *arXiv preprint math/0406446* (2004).

[MNS14] Elchanan Mossel, Joe Neeman, and Allan Sly. "Belief propagation, robust reconstruction and optimal recovery of block models." In: *Conference on Learning Theory*. PMLR. 2014, pp. 356–370.

[MNS15] Elchanan Mossel, Joe Neeman, and Allan Sly. "Reconstruction and estimation in the planted partition model." In: *Probability Theory and Related Fields* 162.3 (2015), pp. 431–461.

[MNS18] Elchanan Mossel, Joe Neeman, and Allan Sly. "A proof of the block model threshold conjecture." In: *Combinatorica* 38.3 (2018), pp. 665–708.

[Muc+10] Peter J Mucha, Thomas Richardson, Kevin Macon, Mason A Porter, and Jukka-Pekka Onnela. "Community structure in time-dependent, multiscale, and multiplex networks." In: *science* 328.5980 (2010), pp. 876–878.

[ML09] Marius Muja and David G Lowe. "Fast approximate nearest neighbors with automatic algorithm configuration." In: *VISAPP (1)* 2.331-340 (2009), p. 2.

[MWJ13] Kevin Murphy, Yair Weiss, and Michael I Jordan. "Loopy belief propagation for approximate inference: An empirical study." In: *arXiv preprint arXiv:1301.6725* (2013).

[New03] Mark EJ Newman. "The structure and function of complex networks." In: *SIAM review* 45.2 (2003), pp. 167–256.

[New04a] Mark EJ Newman. "Detecting community structure in networks." In: *The European physical journal B* 38.2 (2004), pp. 321–330.

[New04b] Mark EJ Newman. "Fast algorithm for detecting community structure in networks." In: *Physical review E* 69.6 (2004), p. 066133.

[New06] Mark EJ Newman. "Modularity and community structure in networks." In: *Proceedings of the national academy of sciences* 103.23 (2006), pp. 8577–8582.

[New16] Mark EJ Newman. "Equivalence between modularity optimization and maximum likelihood methods for community detection." In: *Physical Review E* 94.5 (2016), p. 052315.

[NFB02] Mark EJ Newman, Stephanie Forrest, and Justin Balthrop. "Email networks and the spread of computer viruses." In: *Physical Review E* 66.3 (2002), p. 035101.

[NG04] Mark EJ Newman and Michelle Girvan. "Finding and evaluating community structure in networks." In: *Physical review E* 69.2 (2004), p. 026113.

[NJW01a] Andrew Ng, Michael Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm." In: *Advances in neural information processing systems* 14 (2001), pp. 849–856.

[NJW01b] Andrew Ng, Michael Jordan, and Yair Weiss. "On spectral clustering: Analysis and an algorithm." In: *Advances in neural information processing systems* 14 (2001), pp. 849–856.

[NS13] SS Nishana and Subu Surendran. "Graph embedding and dimensionality reduction-a survey." In: *International Journal of Computer Science & Engineering Technology (IJCSET)* 4.1 (2013), pp. 29–34.

[Nis81] Hidetoshi Nishimori. "Internal energy, specific heat and correlation function of the bond-random Ising model." In: *Progress of Theoretical Physics* 66.4 (1981), pp. 1169–1181.

[NS01] Hidetoshi Nishimori and David Sherrington. "Absence of replica symmetry breaking in a region of the phase diagram of the Ising spin glass." In: *AIP Conference Proceedings*. Vol. 553. 1. American Institute of Physics. 2001, pp. 67–72.

[Ons36] Lars Onsager. "Electric moments of molecules in liquids." In: *Journal of the American Chemical Society* 58.8 (1936), pp. 1486–1493.

[Ons44] Lars Onsager. "Crystal statistics. I. A two-dimensional model with an order-disorder transition." In: *Physical Review* 65.3-4 (1944), p. 117.

[OS01] Manfred Opper and David Saad. *Advanced mean field methods: Theory and practice*. MIT press, 2001.

[PS11] Raj Kumar Pan and Jari Saramäki. "Path lengths, correlations, and centrality in temporal networks." In: *Physical Review E* 84.1 (2011), p. 016105.

[Par88] Giorgio Parisi. *Statistical field theory*. Addison-Wesley, 1988.

[Pea14] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier, 2014.

[Pea01] Karl Pearson. "LIII. On lines and planes of closest fit to systems of points in space." In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572.

[Ped+11] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python." In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Pei36] RE Peierls. "On Ising's ferromagnet model." In: *Proc. Camb. Phil. Soc.* Vol. 32. 1936, pp. 477–481.

[Pei14a] Tiago P Peixoto. "Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models." In: *Physical Review E* 89.1 (2014), p. 012804.

[Pei14b] Tiago P Peixoto. "The graph-tool python library." In: *figshare* (2014).

[Pei19] Tiago P Peixoto. "Bayesian stochastic blockmodeling." In: *Advances in network clustering and blockmodeling* (2019), pp. 289–332.

[Pel05] Alessandro Pelizzola. "Cluster variation method in statistical physics and probabilistic graphical models." In: *Journal of Physics A: Mathematical and General* 38.33 (2005), R309.

[PZ+19] Marianna Pensky, Teng Zhang, et al. "Spectral clustering in the dynamic stochastic block model." In: *Electronic Journal of Statistics* 13.1 (2019), pp. 678–709.

[PSS10] Teresa M Przytycka, Mona Singh, and Donna K Slonim. "Toward the dynamic interactome: it's about time." In: *Briefings in bioinformatics* 11.1 (2010), pp. 15–29.

[QR13] Tai Qin and Karl Rohe. "Regularized spectral clustering under the degree-corrected stochastic blockmodel." In: *arXiv preprint arXiv:1309.4111* (2013).

[Qin+16] Xuanmei Qin, Weidi Dai, Pengfei Jiao, Wenjun Wang, and Ning Yuan. "A multi-similarity spectral clustering method for community detection in dynamic networks." In: *Scientific reports* 6.1 (2016), pp. 1–11.

[Rad+04] Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. "Defining and identifying communities in networks." In: *Proceedings of the national academy of sciences* 101.9 (2004), pp. 2658–2663.

[RM15] Dinesh Ramasamy and Upamanyu Madhow. "Compressive spectral embedding: sidestepping the SVD." In: *arXiv preprint arXiv:1509.08360* (2015).

[RL08] Joerg Reichardt and Michele Leone. "(Un) detectable cluster structure in sparse networks." In: *Physical review letters* 101.7 (2008), p. 078701.

[RB69] Franz Rellich and Joseph Berkowitz. *Perturbation theory of eigenvalue problems*. CRC Press, 1969.

[Ris+01] Irina Rish et al. "An empirical study of the naive Bayes classifier." In: *IJCAI 2001 workshop on empirical methods in artificial intelligence*. Vol. 3. 22. 2001, pp. 41–46.

[Rog+08] Tim Rogers, Isaac Pérez Castillo, Reimer Kühn, and Koujin Takeda. "Cavity approach to the spectral density of sparse symmetric random matrices." In: *Physical Review E* 78.3 (2008), p. 031116.

[RCY+11] Karl Rohe, Sourav Chatterjee, Bin Yu, et al. "Spectral clustering and the high-dimensional stochastic blockmodel." In: *Annals of Statistics* 39.4 (2011), pp. 1878–1915.

[RM05] Lior Rokach and Oded Maimon. "Clustering methods." In: *Data mining and knowledge discovery handbook*. Springer, 2005, pp. 321–352.

[RC18] Giulio Rossetti and Rémy Cazabet. "Community discovery in dynamic networks: a survey." In: *ACM Computing Surveys (CSUR)* 51.2 (2018), pp. 1–37.

[RB10] Martin Rosvall and Carl T Bergstrom. "Mapping change in large networks." In: *PloS one* 5.1 (2010), e8694.

[Rou87] Peter J Rousseeuw. "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis." In: *Journal of computational and applied mathematics* 20 (1987), pp. 53–65.

[Saa92] Youcef Saad. *Numerical methods for large eigenvalue problems*. Manchester University Press, 1992.

[SKZ14] Alaa Saade, Florent Krzakala, and Lenka Zdeborová. "Spectral Clustering of graphs with the Bethe Hessian." In: *NIPS*. 2014.

[Saa+16] Alaa Saade, Marc Lelarge, Florent Krzakala, and Lenka Zdeborová. "Clustering from sparse pairwise measurements." In: *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE. 2016, pp. 780–784.

[Sal11] Justin Salez. "Some implications of local weak convergence for sparse random graphs." PhD thesis. Université Pierre et Marie Curie-Paris VI; Ecole Normale Supérieure de Paris …, 2011.

[SMM14] Iwao Sato, Hideo Mitsuhashi, and Hideaki Morita. "A matrix-weighted zeta function of a graph." In: *Linear and Multilinear Algebra* 62.1 (2014), pp. 114–125.

[Sau+06] Lawrence K Saul, Kilian Q Weinberger, Fei Sha, Jihun Ham, and Daniel D Lee. "Spectral methods for dimensionality reduction." In: *Semi-supervised learning* 3 (2006).

[Sch07] Satu Elisa Schaeffer. "Graph clustering." In: *Computer science review* 1.1 (2007), pp. 27–64.

[SMR08] Hinrich Schütze, Christopher D Manning, and Prabhakar Raghavan. *Introduction to information retrieval*. Vol. 39. Cambridge University Press Cambridge, 2008.

[Sco88] John Scott. "Social network analysis." In: *Sociology* 22.1 (1988), pp. 109–127.

[SM00] Jianbo Shi and Jitendra Malik. "Normalized cuts and image segmentation." In: *IEEE Transactions on pattern analysis and machine intelligence* 22.8 (2000), pp. 888–905.

[Siloo] John R Silvester. "Determinants of block matrices." In: *The Mathematical Gazette* 84.501 (2000), pp. 460–467.

[Sim91] Herbert A Simon. "The architecture of complexity." In: *Facets of systems science*. Springer, 1991, pp. 457–476.

[Ste+11] Juliette Stehlé, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-François Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Régis, Bruno Lina, et al. "High-resolution measurements of face-to-face contact patterns in a primary school." In: *PloS one* 6.8 (2011), e23176.

[SN13] Daniel L Stein and Charles M Newman. *Spin glasses and complexity*. Vol. 4. Princeton University Press, 2013.

[SM20] Ludovic Stephan and Laurent Massoulié. "Non-backtracking spectra of weighted inhomogeneous random graphs." In: *arXiv preprint arXiv:2004.07408* (2020).

[TS92] Jean Tague-Sutcliffe. "An introduction to informetrics." In: *Information processing & management* 28.1 (1992), pp. 1–3.

[Ter10] Audrey Terras. *Zeta functions of graphs: a stroll through the garden*. Vol. 128. Cambridge University Press, 2010.

[Tho86] DJ Thouless. "Spin-glass on a Bethe lattice." In: *Physical review letters* 56.10 (1986), p. 1082.

[TAP77] David J Thouless, Philip W Anderson, and Robert G Palmer. "Solution of 'solvable model of a spin glass'." In: *Philosophical Magazine* 35.3 (1977), pp. 593–601.

[TM01] Françoise Tisseur and Karl Meerbergen. "The quadratic eigenvalue problem." In: *SIAM review* 43.2 (2001), pp. 235–286.

[TPV09] Francesco Parisen Toldin, Andrea Pelissetto, and Ettore Vicari. "Strong-disorder paramagnetic-ferromagnetic fixed point in the square-lattice±J Ising model." In: *Journal of Statistical Physics* 135.5 (2009), pp. 1039–1061.

[TWVE19] Vincent A Traag, Ludo Waltman, and Nees Jan Van Eck. "From Louvain to Leiden: guaranteeing well-connected communities." In: *Scientific reports* 9.1 (2019), pp. 1–12.

[TL20] Nicolas Tremblay and Andreas Loukas. "Approximating spectral clustering via sampling: a review." In: *Sampling Techniques for Supervised or Unsupervised Tasks* (2020), pp. 129–183.

[Tre+16a] Nicolas Tremblay, Gilles Puy, Pierre Borgnat, Rémi Gribonval, and Pierre Vandergheynst. "Accelerated spectral clustering using graph filtering of random signals." In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Ieee. 2016, pp. 4094–4098.

[Tre+16b] Nicolas Tremblay, Gilles Puy, Rémi Gribonval, and Pierre Vandergheynst. "Compressive spectral clustering." In: *International conference on machine learning*. PMLR. 2016, pp. 1002–1011.

[Usa+19] Muhammad Usama, Junaid Qadir, Aunn Raza, Hunain Arif, Kok-Lim Alvin Yau, Yehia Elkhatib, Amir Hussain, and Ala Al-Fuqaha. "Unsupervised machine learning for networking: Techniques, applications and research challenges." In: *IEEE Access* 7 (2019), pp. 65579–65615.

[VL91] Jan Van Leeuwen. *Handbook of theoretical computer science (vol. A) algorithms and complexity*. Mit Press, 1991.

[VM10] Piet Van Mieghem. *Graph spectra for complex networks*. Cambridge University Press, 2010.

[Vez03] A Vezzani. "Spontaneous magnetization of the Ising model on the Sierpinski carpet fractal, a rigorous result." In: *Journal of Physics A: Mathematical and General* 36.6 (2003), p. 1593.

[VL07] Ulrike Von Luxburg. "A tutorial on spectral clustering." In: *Statistics and computing* 17.4 (2007), pp. 395–416.

[WJ08] Martin J Wainwright and Michael Irwin Jordan. *Graphical models, exponential families, and variational inference*. Now Publishers Inc, 2008.

[WW17] Ke Wang and Philip Matchett Wood. "Limiting empirical spectral distribution for the non-backtracking matrix of an Erdos Renyi random graph." In: *arXiv preprint arXiv:1710.11015* (2017).

[WF+94] Stanley Wasserman, Katherine Faust, et al. "Social network analysis: Methods and applications." In: (1994).

[WF10] Yusuke Watanabe and Kenji Fukumizu. "Graph zeta function in the Bethe free energy and loopy belief propagation." In: *arXiv preprint arXiv:1002.3307* (2010).

[WF11] Yusuke Watanabe and Kenji Fukumizu. "Loopy belief propagation, Bethe free energy and graph zeta function." In: *arXiv preprint arXiv:1103.0605* (2011).

[WS98] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world'networks." In: *nature* 393.6684 (1998), pp. 440–442.

[WK18] Sławomir T Wierzchoń and Mieczysław A Kłopotek. *Modern algorithms of cluster analysis*. Springer, 2018.

[Wig58] Eugene P Wigner. "On the distribution of the roots of certain symmetric matrices." In: *Annals of Mathematics* (1958), pp. 325–327.

[XKS13] Jierui Xie, Stephen Kelley, and Boleslaw K Szymanski. "Overlapping community detection in networks: The state-of-the-art and comparative study." In: *Acm computing surveys (csur)* 45.4 (2013), pp. 1–35.

[XKHI14] Kevin S Xu, Mark Kliger, and Alfred O Hero III. "Adaptive evolutionary clustering." In: *Data Mining and Knowledge Discovery* 28.2 (2014), pp. 304–336.

[Xu15] Kevin Xu. "Stochastic block transition models for dynamic networks." In: *Artificial Intelligence and Statistics*. PMLR. 2015, pp. 1079–1087.

[XW05] Rui Xu and Donald Wunsch. "Survey of clustering algorithms." In: *IEEE Transactions on neural networks* 16.3 (2005), pp. 645–678.

[YL15] Jaewon Yang and Jure Leskovec. "Defining and evaluating network communities based on ground-truth." In: *Knowledge and Information Systems* 42.1 (2015), pp. 181–213.

[YFW+00] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. "Generalized belief propagation." In: *NIPS*. Vol. 13. 2000, pp. 689–695.

[Zac77] Wayne W Zachary. "An information flow model for conflict and fission in small groups." In: *Journal of anthropological research* 33.4 (1977), pp. 452–473.

[Zar+20] Tayeb Zarrouk, Romain Couillet, Florent Chatelain, and Nicolas Le Bihan. "Performance-complexity trade-off in large dimensional statistics." In: *2020 IEEE 30th International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE. 2020, pp. 1–6.

[ZK16] Lenka Zdeborová and Florent Krzakala. "Statistical physics of inference: Thresholds and algorithms." In: *Advances in Physics* 65.5 (2016), pp. 453–552.

[ZR18] Yilin Zhang and Karl Rohe. "Understanding regularized spectral clustering via graph conductance." In: *arXiv preprint arXiv:1806.01468* (2018).

[Zwe14] Katharina Anna Zweig. *Network Representations of Complex Data.* 2014.

RESUMÉ SUBSTANTIEL

CATÉGORISATION ET APPRENTISSAGE

La catégorisation, c'est-à-dire la capacité de regrouper des objets, est profondément ancrée dans l'intelligence humaine et joue un rôle prépondérant dans l'apprentissage. On pourrait même dire que le critère suivi pour catégoriser les objets fournit précisément une définition du concept abstrait lié aux catégories elles-mêmes [Hul20].

Consciemment ou non, l'intelligence humaine permet à chacun d'entre nous d'effectuer deux opérations de base. La première est d'identifier les caractéristiques informatives des données d'entrée, c'est-à-dire celles qui sont nécessaires et suffisantes pour leur attribuer une étiquette. Ceci est fait en fournissant une représentation de l'objet dans un espace paramétrique, de sorte que des objets similaires dans le monde réel correspondent à des points proches dans l'espace paramétrique. Cette étape est connue sous le nom d'*extraction de caractéristiques*. La deuxième étape consiste à attribuer une étiquette à chaque élément de la base des données, en fonction de sa position dans l'espace des caractéristiques. Selon le problème considéré, cela peut être fait principalement de deux manières. L'une est la classification: il s'agit d'attribuer chaque élément à l'un des groupes possibles; l'autre est le regroupement: dans ce cas, l'objectif est de diviser en groupes plusieurs éléments, en fonction de leur proximité.

L'intérêt porté à l'*extraction de caractéristiques* et à l'étape suivante de *clustering* ou *classification* va au-delà de l'apprentissage humain et est à la base de nombreux algorithmes d'apprentissage automatique [MRT18].

La classification est généralement réalisée à l'aide d'algorithmes *supervisés*, qui constituent une manière très intuitive (en principe) d'envisager l'apprentissage et sont aujourd'hui le plus déployés. Un grand ensemble de données dont les étiquettes sont connues est utilisé pour entraîner un algorithme à générer une règle capable de prédire les étiquettes. Une fois l'apprentissage terminé, la règle peut être utilisée pour classer des nouvelles données. Les réseaux neuronaux font partie de cette classe d'algorithmes. Ils sont nés pour reproduire le processus d'apprentissage qui se produit dans le cerveau humain et constituent aujourd'hui l'état de l'art dans la plupart des domaines en termes de performances. Cette approche présente toutefois des inconvénients: l'étape d'apprentissage peut être coûteuse en termes de temps de calcul, tandis que la création d'une base des données avec leurs étiquettes peut prendre beaucoup de temps et de capacité de stockage.

En ce qui concerne le problème du regroupement, les techniques non supervisées sont généralement préférées dans les cas où la machine ne dispose

d'aucune information supplémentaire en dehors des données à regrouper. L'absence d'"aide extérieure" entrave bien sûr les performances de l'apprentissage non supervisé par rapport à l'apprentissage supervisé, mais les inconvénients introduits par l'étape d'apprentissage sont ici contournés. Les algorithmes d'apprentissage non supervisé profitent en fait de la capacité des machines de traiter d'énormes quantités de données en peu de temps qui leur donnent la possibilité d'exploiter efficacement les relations entre ses éléments.

Bien que les algorithmes d'apprentissage supervisé soient aujourd'hui très populaires en raison de leur capacité à obtenir des performances surhumaines, de sérieuses inquiétudes doivent être soulevées quant à leurs exigences en matière d'échantillons étiquetés et d'entraînement intensif en terme de temps calcul [Usa+19]. En effet, l'augmentation constante de la taille des ensembles de données d'entrée ne peut (et probablement ne *doit*) pas s'accompagner d'une croissance aussi rapide de l'effort de calcul. Dans cette perspective, les techniques non supervisées jouent un rôle fondamental dans l'apprentissage automatique moderne.

MÉTHODES SPECTRALES

LES ÉLÉMENTS DE BASE

Le clustering a plusieurs applications dans différents domaines et est, en général, un problème mal défini [XW05; RM05; Sch07; WK18]. L'une des classes les plus populaires d'algorithmes de clustering, qui est au cœur du travail présenté dans ce manuscrit, est composée des *méthodes spectrales* dont les éléments constitutifs sont décrits comme suit.

La première étape consiste à fournir une représentation sous forme de graphe aux données d'entrée. Ensuite, l'extraction de caractéristiques consiste à mettre en correspondance chaque élément (ou *nœud* du graphe) avec un vecteur de petite dimension [CZC18; NS13]. Cela doit être fait de manière à ce que des nœuds "similaires" soient projetés sur des vecteurs proches. Dans le clustering spectral, la correspondance entre les nœuds et les vecteurs est obtenue en exploitant les vecteurs propres d'une représentation matricielle de graphe adaptée, $M \in \mathbb{R}^{n \times n}$. L'un des principaux avantages du clustering spectral par rapport à d'autres méthodes repose sur ses solides fondements théoriques et son caractère explicable.

L'étape finale consiste à déterminer des frontières dans l'espace en petite dimension pour séparer les points. L'algorithme le plus couramment adopté pour accomplir cette tâche est *k-means* [Mac+67], mais d'autres solutions telles que *k-medoids* [KR09], *expectation maximization* [DLR77] *DBSCAN* [Est+96] et *OPTICS* [Ank+99] sont également des alternatives valables.

Sans doute, la tâche la plus importante dans les méthodes spectrales est d'identifier une représentation matricielle du graphe M telle que ses vecteurs propres soient capables de produire une bonne projection des nœuds dans

l'espace de petite dimension. Certains des choix les plus courants de M sont une combinaison des matrices *degré* et *adjacence* du graphe, que nous allons maintenant définir.

Nous désignons par $A \in \{0,1\}^{n\times n}$ la matrice d'adjacence, définie de telle sorte que $A_{ij} = 1$ si i et j sont connectés et est égale à zéro sinon. La matrice de degré $D \in \mathbb{R}^{n\times n}$ est une matrice diagonale telle que $D_{ij} = \delta_{ij}(A\mathbf{1}_n)_i$.

Certains choix populaires de M sont la matrice d'adjacence A elle-même, la matrice *Laplacienne* $D - A$ et les matrices *Laplacienne normalisée* $L^{\text{sym}} = D^{-1/2}AD^{-1/2}$, $L^{\text{rw}} = D^{-1}A$ [VL07; LR+15], avec des extensions possibles également au cas pondéré dans lequel A est remplacé par $W \in \mathbb{R}^{n\times n}$.

UNE VUE D'ENSEMBLE DES MÉTHODES SPECTRALES

Les méthodes spectrales sont très interdisciplinaire et, afin de décrire au mieux où se situe le travail présenté dans ce manuscrit dans la littérature existante, il faut garder une large perspective sur le problème.

Tout d'abord, comme nous l'avons mentionné précédemment, l'un des principaux intérêts des méthodes spectrales réside dans leur solide base théorique. De puissants outils basés sur la *théorie des matrices aléatoires* permettent de fournir des résultats précis sur les performances et l'applicabilité d'un algorithme spectral. Typiquement, cependant, deux hypothèses importantes doivent être formulées: la taille n de la matrice M doit aller à l'infini et le nombre d'entrées non nulles (que l'on note m) doit aller à l'infini plus vite que n; [CBG+16; AC17, par exemple]. Alors que la première hypothèse consiste simplement à considérer le cadre très intéressant des grands bases de données, d'un point de vue pratique, la seconde hypothèse pose un problème du point de vue du passage à l'échelle des algorithmes de clustering spectral. En effet, le coût numérique du calcul des k vecteurs propres principaux d'une matrice à m entrées non nulles croît comme $O(mk^2)$ [Saa92]. Considérons le cas $m = n^2$, c'est-à-dire que toutes les entrées de la matrice M sont non nulles, pour une petite valeur de k telle que $k = 2$. La complexité est dans ce cas $O(n^2k^2)$ qui est prohibitif pour un ordinateur portable standard lorsque $n > 10^5$ environ.

Le problème de la complexité de calcul du clustering spectral est bien connu dans la littérature et a conduit à plusieurs stratégies d'échantillonnage visant à améliorer l'efficacité du clustering spectral [TL20]. La procédure la plus simple consiste simplement à rendre la matrice M plus parcimonieuse, en mettant certaines de ses entrées à zéro, ce qui améliore significativement la vitesse de calcul de l'algorithme. Considérons en effet une stratégie d'échantillonnage qui conserve en moyenne 10 entrées non nulles dans chaque ligne de M: la limite de calcul de l'ordinateur portable considéré précédemment est maintenant obtenue pour $n = 10^9$ ce qui permet de traiter des bases de données de taille considérable.

Cette procédure d'échantillonnage a cependant un coût. Au-delà du fait qu'en "jetant" certaines entrées (donc de l'information), les performances de clustering peuvent naturellement être plus faibles, la parcimonie est connue depuis longtemps pour être le point faible des méthodes spectrales. En fait, pour de grands niveaux de parcimonie, les choix standards de M ont typiquement des performances faibles par rapport aux algorithmes non-spectraux.

Le problème de la parcimonie n'est pas seulement lié aux procédures de échantillonnage. Une application naturelle du regroupement dans les graphes consiste à diviser les nœuds d'un réseaux social en *communautés*. Les réseaux sociaux réels sont souvent parcimonieux par construction [Bar13] (chaque nœud interagit généralement avec une très petite fraction du total) et, par conséquent, le regroupement spectral peut être peu performant dans les tâches pratiques de détection de communautés. Dans l'ensemble, il s'agit de limitations sévères des méthodes spectrales qui les rendent inadaptées à un large ensemble des bases des données (comme les réseaux sociaux susmentionnés) et qui ne permettent de bonnes performances qu'au prix d'un coût de calcul élevé.

Une percée dans le domaine des méthodes spectrales dans les graphes parcimonieux a toutefois été réalisée par [Krz+13] spécifiquement dans le contexte de la détection de communautés. Le travail de [Krz+13], ainsi que de [Dec+11; SKZ14], est basé sur des intuitions et des méthodes très profondes mais non rigoureuses empruntées à la physique statistique qui ont renouvelé l'intérêt de la communauté scientifique pour le sujet et ont conduit, dans les années suivantes, à formaliser la plupart de leurs résultats de manière mathématiquement rigoureuse. En ce qui concerne la détection de communautés, la principale contribution de [Krz+13; SKZ14] consiste à proposer des matrices alternatives M, telles que les matrices *hessienne de Bethe* et de *non-backtracking*, pour remplacer les choix "classiques".

La hessienne de Bethe joue un rôle fondamental dans ce manuscrit, étant le choix pour M dans tous nos algorithmes proposés. Dans le cas non pondéré, sa définition est la suivante: pour un scalaire $r \geq 1$

$$H_r = (r^2 - 1) I_n + D - rA,$$

où I_n est la matrice identité de taille n. La contribution fondamentale de ce manuscrit est d'argumenter que $M = H_r$ pour une valeur bien choisie de r constitue un choix puissant pour le regroupement avec des méthodes spectrales dans les graphes parcimonieux qui atteint des performances élevées dans les problèmes du monde réel.

Nous allons maintenant détailler nos principaux résultats.

CONTRIBUTIONS

Afin de fournir au lecteur les outils nécessaires pour comprendre au mieux nos contributions, la partie I introduit les résultats techniques exploités dans

le reste. En particulier, le chapitre 1 fournit une introduction aux graphes, ainsi que leurs définitions et propriétés. Ensuite, le chapitre 2 présente les outils fondamentaux de la physique statistique sur lesquels reposent nos principaux résultats. Enfin, le chapitre 3 présente des résultats mathématiques importants sur les propriétés spectrales de deux matrices fondamentales dans ce manuscrit: la matrice de non-backtracking et la hessienne de Bethe susmentionnées.

La partie II est ensuite consacrée à la présentation de nos contributions originales dans le domaine des méthodes spectrales pour la détection de communautés et est ouverte par le chapitre 4 qui fournit une introduction approfondie au problème.

Nous reprenons ici le "flambeau" de nos prédécesseurs et abordons le problème de la manière dont des hypothèses trop simplistes peuvent entraîner de mauvaises performances algorithmiques. Plus précisément, dans [SKZ14], les auteurs ont proposé un algorithme spectral très puissant pour la détection de communautés dans les graphes très parcimoniuex, basé sur la matrice hessienne de Bethe H_r. Leurs résultats théoriques sont formulés sous l'hypothèse que la matrice A est générée à partir du *modèle stochastique par blocs*, selon lequel chaque nœud du graphe a approximativement le même nombre de connections. Cependant, les réseaux sociaux réels (dans lesquels la détection des communautés est de la plus haute importance) sont connus pour être typiquement hétérogènes, dans le sens où le nombre de connections de chaque nœud varie beaucoup à travers le réseau. En conséquence de cette hypothèse trop simpliste, l'algorithme de [SKZ14] (qui prescrit un choix précis de r) a souvent des performances médiocres par rapport aux algorithmes classiques de clustering spectral.

Nos premiers travaux sont présentés au chapitre 5 et expliquent comment nous avons amélioré [SKZ14], en considérant un modèle génératif plus réaliste, appelé *modèle stochastique par blocs de degrés corrigés* (DCSBM) qui permet en effet de tenir compte de l'hétérogénéité du graphe. Le premier de ces travaux est [DC19]

> **LD**, Romain Couillet: *Community detection in sparse realistic graphs: Improving the Bethe-Hessian* in ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (2019)

dans lequel nous avons proposé (pour $k = 2$ classes de taille égale) un choix alternatif de r à celui de [SKZ14], capable de fournir un clustering performant sur des graphes hétérogènes. La paramétrisation r proposée nécessite cependant la connaissance de certains des paramètres du modèle génératif du graphe qui ne peuvent être donnés pour acquis.

Avec notre contribution suivante [DCT19]

> **LD**, Romain Couillet, Nicolas Tremblay: *Revisiting the Bethe-Hessian: Improved Community Detection in Sparse Heterogeneous Graphs* in NeurIPS Advances in Neural Information Processing Systems 32 (2019)

nous avons répondu à certaines des principales questions posé en [DC19] . Premièrement, nous avons étendu nos résultats au contexte de $k \geq 2$ classes de tailles arbitraires, en montrant qu'une séquence de paramètres r correctement choisis doit être sélectionnée pour obtenir un algorithme spectral efficace sur des graphes hétérogènes. Deuxièmement, nous avons montré que ces paramètres r peuvent être estimés de manière non supervisée. Pour ce faire, nous avons dévoilé une propriété imprévue des valeurs propres de la matrice de non-backtracking B utilisée dans [Krz+13] et exploité la relation connue entre B et H_r donnée par la formule d'Ihara-Bass [Ter10]. Enfin, à l'aide de plusieurs arguments et de simulations numériques, nous affirmons que la paramétrisation que nous proposons est *optimale*, en ce sens qu'aucune autre paramétrisation ne devrait être plus performante sur les graphes générés par le DCSBM.

Le fait de disposer d'un moyen pratique d'estimer les valeurs optimales de r de manière non supervisée nous a permis de concevoir et de tester un algorithme pratique de détection de communautés. Les résultats confirment que la paramétrisation améliorée atteint une performance systématiquement plus élevée par rapport à [SKZ14] sur tous les ensembles de données testés.

Dans la littérature, les contributions de la communauté des physiciens ne sont pas les seules à avoir conduit à des algorithmes spectrales très efficaces sur les graphes parcimonieux. En fait, une ligne de travail notable et indépendante, avec une approche ancrée dans les statistiques a proposé l'utilisation de nouvelles matrices *régularisées*. Dans ce resumé nous ne mentionnerons que l'une de ces matrices, $L_\tau^{\text{sym}} = D_\tau^{-1/2} A D_\tau^{-1/2}$, proposée dans [QR13] avec $D_\tau = D + \tau I_n$ pour τ égal au degré moyen du graphe. Cet algorithme est *de facto* l'état de l'art pour la détection de communautés avec des méthodes spectrales sur des graphes réels. Dans le chapitre 6, nous présentons nos travaux relatifs à L_τ^{sym}, commencés dans [DCT20c]

> **LD**, Romain Couillet, Nicolas Tremblay: *Optimal Laplacian regularization for sparse spectral community detection* in ICASSP IEEE International Conference on Acoustics, Speech and Signal Processing (2020)

où nous avons montré qu'il existe une relation claire entre la matrice hessienne de Bethe H_r et le *Laplacien régularisé* L_τ^{sym}. En outre, sur la base de nos résultats dans [DC19; DCT19] , nous avons montré que le choix de τ dans [QR13] est sous-optimal. La régularisation optimale du Laplacien est obtenue avec un ensemble de τ étroitement lié aux r à adopter pour H_r et qui peut être estimé à partir du graphe de manière non supervisée. D'ailleurs, certains des résultats présentés dans les chapitres 5, 6 sont tirés de [DCT20a]

> **LD**, Romain Couillet, Nicolas Tremblay: *A unified framework for spectral clustering in sparse graphs* accepted to Journal of machine learning research

dans lequel nous avons résumé nos résultats dans un *cadre unifié* en faisant des déclarations formelles de nos conjectures et en fournissant des preuves rigoureuses pour une partie d'entre elles. Une autre contribution importante

de [DCT20a] est au centre du chapitre 7 et concerne l'applicabilité de nos algorithmes proposés sur des graphes réels pratiques ainsi que leur mise en œuvre efficace. Nous proposons ici l'Algorithme 7.1 pour la détection de communautés et menons une étude systématique sur des ensembles de données réelles montrant l'efficacité de notre méthode pour des tâches pratiques. De plus, nous avons également publié une implémentation efficace en langage `Julia` de l'Algorithme 7.1 pour la détection de communautés, appelée `CoDeBetHe.jl`. (**Co**mmunity **De**tection with the **Bet**he **He**ssian).

Le chapitre 8 clôt finalement la partie II en partant des résultats posés dans [DCT20a] pour fournir une vue unifiée de plusieurs méthodes de clustering spectral qui ont jusqu'à présent été traitées indépendamment. Dans ce chapitre, en fait, nous montrons clairement que l'algorithme que nous proposons est *adaptatif* à la dureté de la tâche de clustering et que les méthodes courantes de l'état de l'art dans les régimes parcimonieux et denses peuvent être comprises comme des cas limites (sous-optimales) de l'Algorithme 7.1.

Ces résultats posent des bases méthodologiques solides qui nous ont conduit à travailler sur les extensions de nos travaux à des contextes plus généraux que la détection de communautés, traitées dans la partie III. Le premier problème que nous avons considéré concerne la *detection des communautés dans des graphes dynamiques* (DCD), qui enrichit la description en tenant compte de la nature dynamique des graphes du monde réel. Dans [DCT20b]

> **LD**, Romain Couillet, Nicolas Tremblay: *Community detection in sparse time-evolving graphs with a dynamical Bethe-Hessian* in NeurIPS Advances in Neural Information Processing Systems 33 (2020)

nous avons considéré une séquence d'instantanés de graphes à différents moments avec l'hypothèse que les étiquettes des communautés peuvent changer dans le temps, tout en ayant une corrélation positive. Nous avons développé un nouvel algorithme spectral basé sur la matrice hessienne de Bethe dynamique, capable d'exploiter l'information provenant de ces corrélations et qui surpasse largement les concurrents de l'état de l'art. Les résultats de [DCT20b] sont traités dans le chapitre 9 et l'algorithme correspondant fait également partie du paquet `CoDeBetHe.jl`.

Enfin, les travaux de [DCT21]

> **LD**, Romain Couillet, Nicolas Tremblay: *Nishimori meets Bethe: a spectral method for node classification in sparse weighted graphs* in Journal of Statistical Mechanics: Theory and Experiment

sont présentés dans le chapitre 10 et traitent de l'extension de nos travaux au cadre de la sparsification matricielle pour améliorer l'efficacité des méthodes spectrales, tout en conservant des performances élevées. En testant l'algorithme que nous proposons sur un problème prototypique de l'apprentissage automatique d'images de chiens et de chats, nous avons montré qu'une précision presque parfaite peut être obtenue même pour des niveaux de sparsification très élevés, ce qui rend notre algorithme adapté au regroupement

de très grands ensembles de données. Le résultat principal de [DCT21] , cependant, est profondément lié à la physique statistique, décrivant une relation explicite entre l'approximation de Bethe (à partir de laquelle la matrice hessien de Bethe est dérivée) et la *température de Nishimori*. [Nis81], d'une importance fondamentale en physique des *verres de spin* et en inférence optimale bayésienne [Iba99].

Ensemble, ces travaux offrent une nouvelle vision et des algorithmes adaptatifs robustes, reliant la physique statistique et les méthodes spectrales les plus conventionnelles dans des graphes parcimonieux, hétérogènes et dynamiques.

APPENDICES

A

DCSBM PERCOLATION THRESHOLD

In this appendix we prove Theorem 1.1 which provides a necessary and sufficient condition for the existence of a giant component in a multi-class (with unequal sizes) DCSBM. This theorem is an application of [BJR07, Theorem 3.1] to the present DCSBM setting. For convenience, we recall here the formulation of Theorem 1.1.

Theorem A.1 (Percolation threshold in DCSBM, [DCT20a])**.** *Consider a graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ generated according to the DCSBM procedure as in Definition 1.11. Let $\Pi = \mathrm{diag}(\pi_1,\dots,\pi_k)$ and assume that the constant c is the largest eigenvalue of $C\Pi$ with eigenvector $\mathbf{1}_k$. Then, for all large n, with high probability, $\mathcal{G}(\mathcal{V},\mathcal{E})$ has a giant component if and only if $c\Phi > 1$.*

Proof. In our proof we will use [BJR07, Theorem 3.1] which requires to write the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ as a graphon. Let us define the "node" variable $\boldsymbol{x} = \theta\boldsymbol{v} \in \mathcal{S}$, with probability distribution $\mu(\boldsymbol{x})$, where

- $\theta \in \Theta$ is a scalar that distributed according to ν.
- $\boldsymbol{v} \in \{\boldsymbol{e}_1,\dots,\boldsymbol{e}_k\}$ with $\boldsymbol{e}_p \in \mathbb{R}^k$ satisfying $e_{p,\ell} = \delta_{\ell p}$; $\boldsymbol{v}$ is a k-dimensional random vector with law $\mathbb{P}(\boldsymbol{v} = \boldsymbol{e}_p) = \pi_p$; that is, $\boldsymbol{v}$ encodes the class to which node $\boldsymbol{x}$ belongs.
- The probability density of $\boldsymbol{x}$ is denoted by $\mu(\boldsymbol{x})$ and is defined on $\mathcal{S} = \cup_{\ell=1}^{k}\mathcal{S}_\ell$ where $\mathcal{S}_\ell = [\theta_{\min}\boldsymbol{e}_\ell, \theta_{\max}\boldsymbol{e}_\ell]$. The measure $\mu(\boldsymbol{x})$ is equal to $\pi_\ell\nu(\theta)$ if $\boldsymbol{x} \in \mathcal{S}_\ell$ and it is indeed normalized:

$$\int_{\mathcal{S}} d\mu(\boldsymbol{x}) = \sum_{\ell=1}^{k} \pi_\ell \int_{\Theta} d\nu(\theta) = \sum_{\ell=1}^{k} \pi_\ell = 1.$$

Given two such nodes, we next define the *graphical kernel* $\kappa(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{x}^T C\boldsymbol{y}$, satisfying the three conditions of [BJR07, Definition 2.7]. With these notations at hand, the generative procedure of the DCSBM under consideration is equivalent to: drawing n independent values $\{\boldsymbol{x}_i\}_{i=1,\dots,n}$ from $\mu(\cdot)$ and then generating the edges independently according to $p_{ij} = \min\left(\frac{\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)}{n}, 1\right)$; we thus fall precisely under Bollobas' framework.

In order to use the core argument of [BJR07, Theorem 3.1], we still need to define the linear operator T_κ, an operator on $f : \mathbb{R}^k \to \mathbb{R}$, endowed with the norm $\|f\|_2^2 = \int_{\mathcal{S}} f^2(\boldsymbol{x})\, d\mu(\boldsymbol{x})$, as:

$$\forall\, \boldsymbol{x} \in \mathcal{S}, \quad (T_k f)(\boldsymbol{x}) = \int_{\mathcal{S}} \kappa(\boldsymbol{x},\boldsymbol{y}) f(\boldsymbol{y})\, d\mu(\boldsymbol{y})$$

The square 2-norm of this operator, $\|T_\kappa\|_2$, reads

$$\|T_\kappa\|_2^2 = \sup_{\|f\|_2 \leq 1} \int_{\mathcal{S}} (T_\kappa f)^2(\boldsymbol{x})\, d\mu(\boldsymbol{x}). \tag{A.1}$$

According to [BJR07, Theorem 3.1], with high probability, a giant component appears in $\mathcal{G}(\mathcal{V}, \mathcal{E})$ if and only if $\|T_\kappa\| > 1$. We thus are left to evaluating $\|T_\kappa\|$ for the kernel $\boldsymbol{x}^T C \boldsymbol{y}$: we will show that it equals $c\Phi$, first by showing that $\|T_\kappa\| \leq c\Phi$ and then finding a particular function f for which the bound is attained.

Let f be a function from $\mathbb{R}^k$ to $\mathbb{R}$. We introduce its associated vector $\boldsymbol{\omega}_f \in \mathbb{R}^k$ as follows.

$$\begin{aligned}(T_\kappa f)(\boldsymbol{x}) &= \int_{\mathcal{S}} \kappa(\boldsymbol{x}, \boldsymbol{y}) f(\boldsymbol{y})\, d\mu(\boldsymbol{y}) = \int_{\mathcal{S}} \boldsymbol{x}^T C \boldsymbol{y} f(\boldsymbol{y})\, d\mu(\boldsymbol{y}) \\ &= \boldsymbol{x}^T C \sum_{\ell=1}^{k} \pi_\ell \boldsymbol{e}_\ell \int_\Theta \theta\, f(\theta \boldsymbol{e}_\ell)\, d\nu(\theta) \equiv \boldsymbol{x}^T C \Pi \boldsymbol{\omega}_f, \end{aligned} \tag{A.2}$$

where $\forall\, \ell,\ \omega_{f,\ell} = \int_\Theta \theta\, f(\theta \boldsymbol{e}_\ell)\, d\nu(\theta)$. From Equation (A.2), one thus has:

$$\begin{aligned}\int_{\mathcal{S}} (T_\kappa f)^2(\boldsymbol{x}) d\mu(\boldsymbol{x}) &= \int_{\mathcal{S}} \boldsymbol{\omega}_f^T \Pi C \boldsymbol{x} \boldsymbol{x}^T C \Pi \boldsymbol{\omega}_f\, d\mu(\boldsymbol{x}) \\ &= \Phi\, \boldsymbol{\omega}_f^T \Pi C \left(\sum_{\ell=1}^{k} \pi_\ell \boldsymbol{e}_\ell \boldsymbol{e}_\ell^T \right) C \Pi \boldsymbol{\omega}_f \\ &= \Phi\, \boldsymbol{\omega}_f^T \Pi C \Pi C \Pi \boldsymbol{\omega}_f. \end{aligned} \tag{A.3}$$

Injecting the result of Equation (A.3) into Equation (A.1) we obtain

$$\begin{aligned}\|T_\kappa\|_2^2 &= \Phi \sup_{\|f\|_2 \leq 1} \boldsymbol{\omega}_f^T \Pi C \Pi C \Pi \boldsymbol{\omega}_f = \Phi \sup_{\|f\|_2 \leq 1} \frac{\boldsymbol{\omega}_f^T \Pi C \Pi C \Pi \boldsymbol{\omega}_f}{\boldsymbol{\omega}_f^T \Pi \boldsymbol{\omega}_f} \cdot \boldsymbol{\omega}_f^T \Pi \boldsymbol{\omega}_f \\ &\leq \Phi \sup_{\|f\|_2 \leq 1} \frac{\boldsymbol{\omega}_f^T \Pi C \Pi C \Pi \boldsymbol{\omega}_f}{\boldsymbol{\omega}_f^T \Pi \boldsymbol{\omega}_f} \cdot \sup_{\|f\|_2 \leq 1} \boldsymbol{\omega}_f^T \Pi \boldsymbol{\omega}_f \end{aligned} \tag{A.4}$$

Analyzing the first element, we can write for $\boldsymbol{v}_f = \Pi^{1/2} \boldsymbol{\omega}_f$:

$$\begin{aligned}\sup_{\|f\|_2 \leq 1} \frac{\boldsymbol{\omega}_f^T \Pi C \Pi C \Pi \boldsymbol{\omega}_f}{\boldsymbol{\omega}_f^T \Pi \boldsymbol{\omega}_f} &= \sup_{\|f\|_2 \leq 1} \frac{\boldsymbol{v}_f^T \Pi^{1/2} C \Pi C \Pi^{1/2} \boldsymbol{v}_f}{\boldsymbol{v}_f^T \boldsymbol{v}_f} \\ &\leq \sup_{\boldsymbol{v} \in \mathbb{R}^k} \frac{\boldsymbol{v}^T \Pi^{1/2} C \Pi C \Pi^{1/2} \boldsymbol{v}}{\boldsymbol{v}^T \boldsymbol{v}} \end{aligned}$$

i.e.:

$$\sup_{\|f\|_2 \leq 1} \frac{\boldsymbol{\omega}_f^T \Pi C \Pi C \Pi \boldsymbol{\omega}_f}{\boldsymbol{\omega}_f^T \Pi \boldsymbol{\omega}_f} \leq \lambda_1^{\downarrow}(\Pi^{1/2} C \Pi C \Pi^{1/2}) = \lambda_1^{\downarrow}((C\Pi)^2) = c^2,$$

as, by hypothesis, c is the Perron eigenvalue of $C\Pi$, and as such is larger than the modulus of any other eigenvalue: $\lambda_1^{\downarrow}\left((C\Pi)^2\right)$ is indeed c^2. We can therefore rewrite Equation (A.4) as

$$\|T_\kappa\|_2^2 \leq c^2\Phi \sup_{\|f\|_2\leq 1} \boldsymbol{\omega}_f^T \Pi \boldsymbol{\omega}_f.$$

Analyzing the right-hand side term, we have

$$\boldsymbol{\omega}_f^T \Pi \boldsymbol{\omega}_f = \sum_{\ell=1}^{k} \pi_\ell \omega_{f,\ell}^2 = \sum_{\ell=1}^{k} \pi_\ell \left(\int_\Theta \theta f(\theta \boldsymbol{e}_\ell)\, d\nu(\theta) \right)^2$$

which, by Cauchy-Schwartz's inequality is bounded as

$$\left(\int_\Theta \theta f(\theta \boldsymbol{e}_\ell)\, d\nu(\theta) \right)^2 \leq \left(\int_\Theta \theta^2 d\nu(\theta) \right) \left(\int_\Theta f^2(\theta \boldsymbol{e}_\ell)\, d\nu(\theta) \right) \leq \Phi,$$

where in the last step we used the fact that the norm of f is less than or equal to one. We thus obtain that $\sup_{\|f\|_2\leq 1} \boldsymbol{\omega}_f^T \Pi \boldsymbol{\omega}_f \leq \Phi$ and conclude that

$$\|T_\kappa\|_2^2 \leq (c\Phi)^2. \tag{A.5}$$

We are thus left to showing that there exists a function f for which the bound is reached. Let us consider $\bar{f}(\boldsymbol{x}) = \|\boldsymbol{x}\| / \sqrt{\Phi}$. It is easy to check that this function has unit norm

$$\|\bar{f}\|_2^2 = \int_{\mathcal{S}} \left(\frac{\|\boldsymbol{x}\|}{\sqrt{\Phi}} \right)^2 d\mu(\boldsymbol{x}) = \frac{1}{\Phi} \int_{\mathcal{S}} \theta^2 \|\boldsymbol{e}_\ell\|^2\, d\mu(\boldsymbol{x}) = \frac{1}{\Phi} \int_{\mathcal{S}} \theta^2\, d\mu(\boldsymbol{x}) = 1.$$

Furthermore, observe that $\boldsymbol{\omega}_{\bar{f}} = \sqrt{\Phi}\mathbf{1}_k$. Then, we have

$$\begin{aligned} \|T_\kappa\|_2^2 &= \Phi \sup_{\|f\|_2\leq 1} \boldsymbol{\omega}_f^T \Pi C \Pi C \Pi \boldsymbol{\omega}_f \geq \Phi \boldsymbol{\omega}_{\bar{f}}^T \Pi C \Pi C \Pi \boldsymbol{\omega}_{\bar{f}} \\ &\geq \Phi^2 \mathbf{1}_k^T \Pi (C\Pi)^2 \mathbf{1}_k = (c\Phi)^2 \end{aligned} \tag{A.6}$$

as, by hypothesis, we suppose that $C\Pi\mathbf{1}_k = c\mathbf{1}_k$.

Combining Equations (A.5) and (A.6), we find that $\|T_\kappa\| = c\Phi$ so that, from [BJR07, Theorem 3.1], we conclude that the percolation transition is at $c\Phi = 1$ and a giant component exists if and only if $c\Phi > 1$. □

B

OPTIMAL BETHE-HESSIAN

B.1 DEFINITION OF ζ ON THE SPARSE DCSBM

In this appendix we show that for a graph generated from the sparse DCSBM, the two following definitions of $\zeta_p^{(j)}$ (where j is the giant component) are indeed equivalent with high probability for $p \geq 2$ if $\zeta_p^{(j)}$ exists:

$$\zeta_p^{(j)} = \min_{r \geq 1} \{r : \lambda_p^{\uparrow}(H_r^{(j)}) = 0\}, \tag{B.1}$$

$$\zeta_p^{(j)} = \min_{r > 1} \{r : \lambda_p^{\uparrow}(H_r) = 0\}. \tag{B.2}$$

We will further discuss that, with high probability, $\zeta_p^{(j)} > 1$ exists only for the giant component.

This appendix is structured as follows. First we enunciate two lemmas and one corollary to state that when an arbitrary connected graph has at most one cycle, the spectrum of its associated non-backtracking matrix B does not have real eigenvalues that are larger than one in modulus.

We then proceed by arguing that with high probability, for large DCSBM graphs, the small connected components do not have more than one cycle, implying that all the real eigenvalues of B that are larger than one in modulus come from the giant component. Finally we show that this last statement implies that the two definitions of $\zeta_p^{(j)}$ given in Equations (B.1,B.2) are indeed equivalent with high probability. The proofs of the two lemmas and the corollary close Appendix B.1.

Lemma B.1 (Spectrum of B on a tree)**.** *Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a tree. Then, all the eigenvalues of its associated non-backtracking matrix are equal to zero.*

From Lemma B.1 we conclude that non-zero eigenvalues of B imply the existence of cycles in $\mathcal{G}(\mathcal{V}, \mathcal{E})$ which must exist to define the ζ_p's. The second lemma considers the effect of adding a node to an arbitrary graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$.

Lemma B.2 (Spectrum of $\mathcal{G}$ plus a node)**.** *Let $\mathcal{G}'$ be a graph obtained by adding one node and one edge to the graph $\mathcal{G}$, i.e. for $i \notin \mathcal{V}$ and an arbitrary $j \in \mathcal{V}$, $\mathcal{V}' = \mathcal{V} \cup \{i\}$ and $\mathcal{E}' = \mathcal{E} \cup (ij)$. Then, all the non-zero eigenvalues of $B(\mathcal{G}')$ are equal to the non-zero eigenvalues of $B(\mathcal{G})$, where $B(\mathcal{G})$ is the non-backtracking matrix of $\mathcal{G}$.*

Applying recursively Lemma B.2 one easily obtains that, attaching a tree to a node of $\mathcal{G}(\mathcal{V},\mathcal{E})$, all the non-zero eigenvalues of B are left unchanged. In simple words, the tree part of a graph is "invisible" to the spectrum of B and contributes to it with null eigenvalues. The last property is a Corollary of Lemma B.2 and shows that cycles lead to non-trivial eigenvalues in the spectrum of B.

Corollary B.1 (Spectral radius of a connected graph with $|\mathcal{E}| = |\mathcal{V}|$)**.** *Let $\mathcal{G}(\mathcal{V},\mathcal{E})$ be a connected graph with $|\mathcal{E}| = |\mathcal{V}|$. Then $\lambda_1^{\downarrow|\cdot|}(B) = 1$.*

A graph with $|\mathcal{E}| = |\mathcal{V}|$ is simply obtained adding an edge between two arbitrary nodes of a tree, creating one cycle.

With the results of the former two Lemmas, we proceed to argue that all the eigenvalues of B that are larger than one in modulus come from the giant component. Let $\Lambda(B)$ be the set of all eigenvalues of B. Since the graph is disconnected, $\Lambda(B) = \bigcup_{j=1}^{n_{cc}} \Lambda\left(B^{(j)}\right)$, i.e., each connected component contributes independently to the eigenvalues of B. The expected size of the small connected components grows as $\log(n)$ [BJR07, Thm. 3.12]. We claim that this implies that the probability that a small connected component contains two or more cycles is $o_n(1)$, thus tending to zero as n grows to infinity. Applying Lemma B.1 (for zero cycles) and Corollary B.1 (for one cycle), we conclude that, with high probability all the real eigenvalues of B larger than one in modulus will come from the giant component. From Theorem 2.1, for all the eigenvalues $\lambda \neq \pm 1$, $\lambda \in \Lambda(B) \iff \det[H_\lambda] = 0$. From the definition of $\zeta_{p\geq 2}^{(j)}$ we gave in Equation (5.2), we have that all the $\zeta_p^{(j)} > 1$ are in the spectrum of B. We may conclude that, if $\exists\, j$ such that $\zeta_{p\geq 2}^{(j)} > 1$, then with high probability j is the giant component. We now conclude showing that this statement implies the equivalence between (B.1) and (B.2).

Let $p > 1$, then certainly $\zeta_p^{(j)} > 1$, if it exists. We denote with j the giant component and we want to prove that for $r, p \geq 2$

$$\lambda_p^{\uparrow}(H_r^{(j)}) = \lambda_p^{\uparrow}(H_r) \tag{B.3}$$

Consider $\zeta_p^{(j)} > 1$ such that $\lambda_p^{\uparrow}(H^{(j)}_{\zeta_p^{(j)}}) = 0$, then certainly there exists[1] $q \geq p$ such that $\lambda_q^{\uparrow}(H_{\zeta_p^{(j)}}) = 0$ and so it is enough to show that $p = q$. We proceed with a proof by contradiction. Suppose that $p \neq q$, then there exists $j' \neq j$ such that $\lambda_1^{\uparrow}\left(H^{(j')}_{\zeta_p^{(j)}}\right) \leq 0$. Applying Gershgorin circle theorem [JH85], it is easy to show that for $r > d_{\max} - 1$, the matrix H_r is positive definite. Consequently, there exists $r \geq \zeta_p^{(j)} > 1$ such that $\lambda_1^{\uparrow}(H_r^{(j')}) = 0$. From the Ihara-Bass formula, r is thus in $\Lambda(B)$, so there is a real eigenvalue of B larger than one not coming from the giant component. This is in contradiction with what stated above, thus concluding our argument.

1 As $\Lambda(H_r) = \cup_{j=1}^{n_{cc}} \Lambda\left(H_r^{(j)}\right)$.

We now provide a proof of Lemma B.1, B.2 and Corollary B.1.

Proof of Lemma B.1. Consider a vector $\boldsymbol{g} \in \mathbb{R}^{|\mathcal{E}_{\mathrm{d}}|}$ and define $\boldsymbol{g}^{(m)} \in \mathbb{R}^{|\mathcal{E}_{\mathrm{d}}|}$ as

$$g_{ij}^{(m)} = \sum_{(k\ell)\,:\,d(ij,k\ell)=m} g_{k\ell}. \tag{B.4}$$

The notation $d(ij,k\ell) = m$ indicates that there exists a non-backtracking path from the edge (ij) to the edge $(k\ell)$ of length m. From a straightforward calculation one obtains

$$(B\boldsymbol{g}^{(m)})_{ij} = \sum_{\ell \in \partial j \setminus i} \sum_{(kq):d(j\ell,kq)=m} g_{kq} = g_{ij}^{(m+1)}. \tag{B.5}$$

For all trees, for any two edges we have that $d(ij,k\ell) \leq n-1$, so there is a value of $m_{\mathrm{c}} \leq n-1$, which represents the maximal distance between any two directed edges, such that, for all vectors $\boldsymbol{g}$,

$$B\boldsymbol{g}^{(m_{\mathrm{c}})} = 0. \tag{B.6}$$

This relation comes from the fact that, by definition of m_{c}, no two edges are at a distance equal to $m_{\mathrm{c}}+1$, so $\{(k\ell : d(ij,k\ell) = m_{\mathrm{c}}+1\} = \emptyset$ for any edge (ij). Now, let us consider $\boldsymbol{g}$ to be an eigenvector of B, such that $B\boldsymbol{g} = \lambda \boldsymbol{g}$. Then we can write

$$0 = B\boldsymbol{g}^{(m_{\mathrm{c}})} = B^2\boldsymbol{g}^{(m_{\mathrm{c}}-1)} = \cdots = B^{m_{\mathrm{c}}-1}\boldsymbol{g} = \lambda^{m_{\mathrm{c}}-1}\boldsymbol{g}. \tag{B.7}$$

Thus concluding that any eigenvector $\boldsymbol{g}$ of B is associated to eigenvalue zero. Note that in other words, this means that B is nilpotent. □

Proof of Lemma B.2. Let i be the newly added node and j the node in $\mathcal{V}$ to which i is attached. The matrix $B(\mathcal{G}')$ can be written by adding to the matrix $B(\mathcal{G})$ two rows and two columns corresponding to the directed edges (ij) and (ji). We introduce the notation $\mathbb{I}^{(\bullet j)} \in \mathbb{R}^{|\mathcal{E}_{\mathrm{d}}|}$. Its element-wise definition reads for all $(yx) \in \mathcal{E}_{\mathrm{d}}$ as $\mathbb{I}_{yx}^{(\bullet j)} = \delta_{xj}$. Similarly, we define $\mathbb{I}^{(j\bullet)} \in \mathbb{R}^{|\mathcal{E}_{\mathrm{d}}|}$ with $\mathbb{I}_{yx}^{(j\bullet)} = \delta_{yj}$. Denote with $M = (\mathbb{I}^{(j\bullet)}, \mathbf{0}_{|\mathcal{E}_{\mathrm{d}}|}) \in \{0,1\}^{|\mathcal{E}|\times 2}$ and $M' = (\mathbf{0}_{|\mathcal{E}_{\mathrm{d}}|}, \mathbb{I}^{(\bullet j)}) \in \{0,1\}^{|\mathcal{E}_{\mathrm{d}}|\times 2}$. The matrix $B(\mathcal{G}')$ can be written as:

$$B(\mathcal{G}') = \begin{pmatrix} B(\mathcal{G}) & M' \\ M^T & 0_{2\times 2} \end{pmatrix}$$

We now look for the non-zero eigenvalues of λ of $B(\mathcal{G}')$. Consider $\lambda \in \mathbb{C}^*$. Using a block matrix determinantal formula one has:

$$\det\left(B(\mathcal{G}') - \lambda I_{2|\mathcal{E}'_{\mathrm{d}}|}\right) = \det(-\lambda I_2)\det\left(B(\mathcal{G}) - \lambda I_{|\mathcal{E}_{\mathrm{d}}|} + \frac{1}{\lambda} M'M^T\right).$$

It is straightforward to check that $M'M^T = 0_{|\mathcal{E}_{\mathrm{d}}|\times|\mathcal{E}_{\mathrm{d}}|}$, thus

$$\det\left(B(\mathcal{G}') - \lambda I_{|\mathcal{E}'_{\mathrm{d}}|}\right) = \lambda^2 \det\left(B(\mathcal{G}) - \lambda I_{|\mathcal{E}_{\mathrm{d}}|}\right)$$

A non-zero eigenvalue of $B(\mathcal{G}')$ (cancelling the determinant) is thus necessarily also an eigenvalue of $B(\mathcal{G})$, ending the proof. □

Proof of Corollary B.1. A connected graph with $|\mathcal{E}| = |\mathcal{V}|$ can be obtained by adding an edge between any two nodes of a particular tree defined on $\mathcal{V}$. The graph $\mathcal{G}$ thus contains only one cycle. We now apply Lemma B.2 "backwards", i.e., removing nodes with unitary degree (called *leaves*) from $\mathcal{G}(\mathcal{V},\mathcal{E})$ without affecting the non-zero eigenvalues of B. By iteratively removing all leaves, the graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ reduces to a cycle. We are thus left to prove that $\lambda_1^{\downarrow|\cdot|}(B) = 1$ on a cycle. It is straightforward to check that on any d-regular graph, the vector $\mathbf{1}_{|\mathcal{E}_d|}$ is the Perron-Frobenius eigenvector of B with eigenvalue equal to $d-1$. A cycle is a d-regular graph with $d = 2$, hence the result. □

B.2 THE EIGENVALUES OF T

We here want to show that $\lambda_p^{\downarrow}(T) = \frac{1}{c}\lambda_{p+1}^{\downarrow}(C\Pi)$ for $1 \leq p < k$. Consider the following equivalent identities for the matrix $T = \frac{\Pi C}{c} - \Pi \mathbf{1}_k \mathbf{1}_k^T$:

$$\left(\frac{\Pi C}{c} - \Pi \mathbf{1}_k \mathbf{1}_k^T\right) \boldsymbol{a}_p = \lambda_p^{\downarrow}(T)\boldsymbol{a}_p$$
$$\left(\frac{C\Pi}{c} - \mathbf{1}_k \mathbf{1}_k^T \Pi\right) \boldsymbol{b}_p = \lambda_p^{\downarrow}(T)\boldsymbol{b}_p; \qquad \boldsymbol{b}_p = \Pi^{-1}\boldsymbol{a}_p$$

Since $C\Pi\mathbf{1} = c\mathbf{1}_k$ and $\mathbf{1}_k^T \Pi \mathbf{1}_k = 1$, then $\boldsymbol{b}_k = \mathbf{1}_k$. By introducing $\boldsymbol{d}_p = \Pi^{1/2}\boldsymbol{b}_p$, we can write:

$$\left(\frac{\Pi^{1/2} C \Pi^{1/2}}{c} - \Pi^{1/2}\boldsymbol{b}_k \boldsymbol{b}_k^T \Pi^{1/2}\right) \boldsymbol{d}_p = \lambda_p^{\downarrow}(T)\boldsymbol{d}_p$$
$$\left(\frac{\Pi^{1/2} C \Pi^{1/2}}{c} - \boldsymbol{d}_k \boldsymbol{d}_k^T\right) \boldsymbol{d}_p = \lambda_p^{\downarrow}(T)\boldsymbol{d}_p$$

Given that the $\boldsymbol{d}_k$ are eigenvectors of a symmetric matrix, then $\boldsymbol{d}_p^T \boldsymbol{d}_q = \delta_{pq}$, so, for $p < k$

$$\frac{\Pi^{1/2} C \Pi^{1/2}}{c}\boldsymbol{d}_p = \lambda_p^{\downarrow}(T)\boldsymbol{d}_p$$
$$\frac{C\Pi}{c}\boldsymbol{b}_p = \lambda_p^{\downarrow}(T)\boldsymbol{b}_p$$

From this last equation the result follows directly.

C

SPECTRAL CLUSTERING IN DYNAMICAL GRAPHS

C.1 DETECTABILITY THRESHOLD AT FINITE T

In this appendix we provide the supporting arguments to Claim 9.1.

Recall the Definition 9.4 of the *aggregate graph* $\mathcal{G}$. As a consequence of the sparsity of each $\mathcal{G}_t$, the graph $\mathcal{G}$, obtained by connecting together the same node at successive times is locally tree-like, *i.e.* the local structure of $\mathcal{G}$ around a node $v \in \mathcal{V}$ is the same as that of a Galton-Watson tree $\mathcal{T}(v)$ [DM+10a], rooted at v, designed according to the following procedure:

1. Let $\ell_v \in \{1, 2\}$ be the label of v
2. Next generate its progeny by creating d_s spatial children (*i.e.*, nodes which live at the same time as v), where d_s is a Bernoulli random variable with mean $c\Phi$, and two temporal children (*i.e.*, nodes which are the projection of v at neighbouring times)
3. For each spatial child w, assign the label $\ell_w = \ell_v$ with probability $c_{\text{in}}/(c_{\text{in}} + c_{\text{out}})$ and $\ell_w = 3 - \ell_v$ otherwise; the temporal children keep the same label as v with probability $(1 + \eta)/2$ and change it with probability $(1 - \eta)/2$
4. Each node thus created further generates its own set of offspring, with the only difference that the temporal children only bear one extra temporal child, while spatial children bear two.

In the limit $n, T \to \infty$, for any arbitrary $v \in \mathcal{V}$, the probability distribution around a neighbourhood of v reachable in a finite number of steps is asymptotically the same as $\mathcal{T}(v)$, the Galton-Watson tree rooted at v. The local tree-like structure is preserved for finite T (and $n \to \infty$) but the boundary conditions imposed by $t = 1$ and $t = T$ must be accounted for.

Taking inspiration from the results of [JM+04] on robust reconstruction on trees, the authors of [Gha+16] conjectured a generalization for a multitype branching process, such as just described to construct $\mathcal{T}(v)$. The conjecture of [Gha+16] (which we adapted to the DDCSBM, redefining α, follow-

ing the work of [GLM15]) states that, for $T \to \infty$, community detection is possible if and only if the largest eigenvalue of

$$M_\infty(\alpha, \eta) = \begin{pmatrix} \alpha^2 & 2\eta^2 \\ \alpha^2 & \eta^2 \end{pmatrix} \tag{C.1}$$

is greater than one. Taking a closer look at M_∞, one sees that first row is given by the contribution of spatial children,[1] while the second one from temporal children. For finite T, at each time instant, one can identify three types of edges: spatial edges (connecting nodes in $\mathcal{G}_t$ to nodes in $\mathcal{G}_t$), forward temporal edges (connecting nodes in $\mathcal{G}_t$ to nodes in $\mathcal{G}_{t+1}$) and backwards temporal edges (connecting nodes in $\mathcal{G}_t$ to nodes in $\mathcal{G}_{t-1}$). We construct a matrix $\tilde{M}_T(\alpha, \eta) \in \mathbb{R}^{3T \times 3T}$ identifying the rows and the columns as

$$\{(\text{backwards temporal})_t, (\text{spatial})_t, (\text{forward temporal})_t\}_{t=1,\dots,T}.$$

A $(\text{backwards temporal})_t$ edge goes from a node in $\mathcal{V}_t$ to a node in $\mathcal{V}_{t-1}$ that has, on average, $c\Phi$ spatial children with label correlation equal to $\gamma = (c_{\text{in}} - c_{\text{out}})/(c_{\text{in}} + c_{\text{out}})$ and one backwards temporal child, with label correlation equal to η. Similarly $(\text{spatial})_t$ goes from a node in $\mathcal{V}_t$ to a node in $\mathcal{V}_t$ having $c\Phi$ temporal children and, one forward and one backwards temporal children; finally, $(\text{forward temporal})_t$ goes from $\mathcal{V}_t$ to $\mathcal{V}_{t+1}$ with one forward temporal child and $c\Phi$ spatial children. The entry i, j of $\tilde{M}_T(\alpha, \eta)$ is then set equal to the number of off-springs of type j of a node reached by an edge of type i, multiplied by the square label correlation. As forward temporal edges do not exist for $t = T$ and backwards temporal edges do not exist for $t = 1$, the matrix $\tilde{M}_T(\alpha, \eta) \in \mathbb{R}^{3T \times 3T}$ takes the form

$$\tilde{M}_T(\alpha, \eta) = \begin{pmatrix} \tilde{M}_d^+ & M_+ & 0 & \dots & 0 \\ \tilde{M}_- & M_d & \ddots & \dots & 0 \\ 0 & M_- & \ddots & M_+ & 0 \\ \vdots & \vdots & \ddots & M_d & \tilde{M}_+ \\ 0 & 0 & \dots & M_- & \tilde{M}_d^- \end{pmatrix} \tag{C.2}$$

where

$$M_d = \begin{pmatrix} 0 & 0 & 0 \\ \eta^2 & c\Phi\gamma^2 & \eta^2 \\ 0 & 0 & 0 \end{pmatrix}; \quad M_+ = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & c\Phi\gamma^2 & \eta^2 \end{pmatrix}; \quad M_- = \begin{pmatrix} \eta^2 & c\Phi\gamma^2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$\tilde{M}_d^+ = \begin{pmatrix} 0 & 0 & 0 \\ 0 & c\Phi\gamma^2 & \eta^2 \\ 0 & 0 & 0 \end{pmatrix}; \quad \tilde{M}_+ = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & c\Phi\gamma^2 & 0 \end{pmatrix};$$

$$\tilde{M}_- = \begin{pmatrix} 0 & c\Phi\gamma^2 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}; \quad \tilde{M}_d^- = \begin{pmatrix} 0 & 0 & 0 \\ \eta^2 & c\Phi\gamma^2 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

1 Note that $\alpha^2 = c\Phi\gamma$, where $\gamma = (c_{\text{in}} - c_{\text{out}})/(c_{\text{in}} + c_{\text{out}})$ is the label covariance of neighbouring nodes.

Note that, since the first and the last rows of $\tilde{M}_T(\alpha,\eta)$ only have zero entries, $\tilde{M}_T(\alpha,\eta)$ has the same non-zero eigenvalues as $M_T(\alpha,\eta)$ of Claim 9.1. This also implies that $M_T(\alpha,\eta)$ shares the non-zero eigenvalues of a matrix of size $(3T-2)\times(3T-2)$ as conjectured in [Gha+16].

C.2 SPECTRUM OF $B_{\xi,h}$

In this appendix we provide support to Claim 9.2, studying the spectrum of $B_{\xi,h}$ with a method that can be seen as a generalization of [Krz+13], introduced in Section 3.2.1. We recall that this method consists in identifying a *guess* eigenvector of $B_{\xi,h}$ and determine the conditions under which its expectation is expected to be a good approximation of one the eigenvectors of the dynamical non-backtracking matrix. To do so, we first determine the position of the eigenvalues belonging to the *informative family* (starting from the largest) and then of the *uninformative family*. Secondly, we analyze the variance of the expression of the expected eigenvector and see under what condition the expectation is meaningful. With this result we finally determine the value of $L_{\xi,h}$ (the radius of the bulk of $B_{\xi,h}$).

C.2.1 THE POSITION OF THE INFORMATIVE EIGENVALUES

In this section we determine the position of the informative eigenvalues of $B_{\xi,h}$ with modulus larger than $L_{\xi,h}$. To do so, we first study the largest of them in the limiting case $T\to\infty$, to then extend our findings for finite T to all other eigenvalues.

The limiting case of $T\to\infty$

Consider the aggregate graph $\mathcal{G}$ of Definition 9.4. Let $\omega_{ij}=\xi$ if there exists t such that $(ij)\in\mathcal{E}_{\mathrm{d},t}$ and $\omega_{ij}=h$ otherwise, and let $\boldsymbol{g}^{(r)}\in\mathbb{R}^{|\mathcal{E}_\mathrm{d}|}$, for $r\in\mathbb{N}$, be the vector with entry

$$g_{ij}^{(r)}=\frac{1}{\lambda_{\mathrm{info},1}^{r}}\sum_{(wx)\,:\,\substack{d(jk,wx)=r\\ k\neq i}} W_{(jk)\to(wx)}\sigma_x, \tag{C.3}$$

where $\{(jk)\ :\ d(jk,wx)=r\}$ is the set of directed edges (jk) such that the shortest directed non-backtracking path connecting (jk) to (wx) is of length r, and where $W_{(jk)\to(wx)}$ is the "total weight" of this shortest path defined as the product of each edge weight ω_{ij}, *i.e*,

$$W_{(jk)\to(wx)}=\omega_{(jk)}\omega_{(k\cdot)}\cdots\omega_{(\cdot w)}\omega_{(wx)}.$$

The quantity $\sigma_x\in\{\pm1\}$ takes its value according to the label of node x. The value of $\lambda_{\mathrm{info},1}$ appearing in Equation (C.3) will be chosen in order to enforce

the vector $\boldsymbol{g}^{(r)}$ to be an approximate eigenvector of $B_{\xi,h}$. By the definition of $\boldsymbol{g}^{(r)}$, we find that

$$(B_{\xi,h}\boldsymbol{g}^{(r)})_{ij} = \lambda_{\text{info},1} g_{ij}^{(r+1)}. \tag{C.4}$$

We now analyze this expression exploiting the tree-like approximation elaborated in Appendix C.1. Resuming from this approximation, the expectation of $g_{ij}^{(r)}$ may be written under the following form:

$$\mathbb{E}[g_{ij}^{(r)}] = \frac{1}{\lambda_{\text{info},1}^{r}} \left(c\Phi\gamma\xi\chi_s^{(r-1)} + \phi_i\eta h\chi_t^{(r-1)} \right) \sigma_j. \tag{C.5}$$

Here the first addend is the contribution of the spatial children of j which are on average $c\Phi$ in number, and for each of them the weight of the connecting edge is equal to ξ while the correlation between the labels $\gamma = \mathbb{E}[\sigma_j\sigma_k]$. Each spatial child being at a distance $r-1$ from the target edges – themselves at a distance r from (jk) – contributes to the sum through a term which we denoted $\chi_s^{(r-1)} > 0$. Similarly, the second addend is the contribution of the temporal children which are $\phi_i = 2$ in number if (ij) is a spatial edge or $\phi_i = 1$ if (ij) is a temporal edge; their own contribution is denoted $\chi_t^{(r-1)} > 0$. The correlation of the labels of temporal children is equal to η and the weight of the edges is equal to h. Importantly note that, as a consequence of γ, ξ, η, h being assumed to be all positive, both $\chi_s^{(r)}$ and $\chi_t^{(r)}$ are positive as well.

By recurrence, the values of $\chi_{s/t}^{(r)}$, which we just defined, then undergo the following relation

$$\begin{aligned}
\begin{pmatrix} \chi_s^{(r)} \\ \chi_t^{(r)} \end{pmatrix} &= \begin{pmatrix} c\Phi\gamma\xi & 2\eta h \\ c\Phi\gamma\xi & \eta h \end{pmatrix} \begin{pmatrix} \chi_s^{(r-1)} \\ \chi_t^{(r-1)} \end{pmatrix} = \begin{pmatrix} c\Phi\gamma\xi & 2\eta h \\ c\Phi\gamma\xi & \eta h \end{pmatrix}^r \begin{pmatrix} \chi_s^{(0)} \\ \chi_t^{(0)} \end{pmatrix} \\
&\equiv \left(M_\infty(\sqrt{c\Phi\gamma\xi}, \sqrt{h\eta}) \right)^r \begin{pmatrix} \chi_s^{(0)} \\ \chi_t^{(0)} \end{pmatrix},
\end{aligned}$$

where $M_\infty(\cdot,\cdot)$ is the matrix introduced in Equation (9.3). For simplicity we will denote it as M_∞. For, say, $r \sim \log(n)$, $\chi_{s/t}^{(r)} \approx \left[\lambda_1^{\downarrow|\cdot|}(M_\infty)\right]^r v_{s/t}$, where $\boldsymbol{v} = (v_s, v_t)$ is the eigenvector associated to the eigenvalue of M_∞ of largest amplitude. Equation (C.5) can therefore be further approximated as

$$\mathbb{E}[g_{ij}^{(r)}] = \left(\frac{\lambda_1^{\downarrow|\cdot|}(M_\infty)}{\lambda_{\text{info},1}} \right)^r (c\Phi\lambda\xi v_s + \phi_i\eta h v_t)\, \sigma_j + o\left(\frac{\lambda_1^{\downarrow|\cdot|}(M_\infty)}{\lambda_{\text{info},1}} \right)^r.$$

This expression naturally leads to the choice $\lambda_{\text{info},1} = \lambda_1^{\downarrow|\cdot|}(M_\infty)$ for which $\mathbb{E}[g_{ij}^{(r)}]$ is independent of r, thus turning Equation (C.4) into an approximate eigenvector equation and $\lambda_{\text{info},1}$ into a close approximation of one of the real eigenvalues of $B_{\xi,h}$.

We now extend this result to the case of finite T, and bring further conclusion on all the eigenvalues of $B_{\xi,h}$ belonging to the *informative family*.

The case of finite T

As we discussed already along Appendix C.1, the case of finite T introduces further difficulties due to the time-boundaries $t = 1$ and $t = T$. This being accounted for, when analyzing the contribution of each edge, not only we have to distinguish between spatial and temporal edges, but also to specify the time at which the edge lives. More precisely, suppose that $j \in \mathcal{V}_t$ for $1 \leq t \leq T$. We can rewrite Equation (C.5) as

$$\mathbb{E}[g_{ij}^{(r)}] = \frac{1}{\lambda_{\text{info},1}} \left[c\Phi\gamma\xi\chi_{s,t}^{(r-1)} + (1-\delta_{1,t})\eta h\chi_{b,t}^{(r-1)} + (1-\delta_{T,t})\eta h\chi_{f,t}^{(r-1)} \right],$$

where $\chi_{s,t}^{(\cdot)}, \chi_{b,t}^{(\cdot)}, \chi_{f,t}^{(\cdot)}$ are respectively the contributions to the of a spatial, a backwards temporal and a forward temporal child of a node $j \in \mathcal{V}_t$. The relation between all the χ's directly unfolds from the branching process at finite T that we already discussed in Appendix C.1. More precisely, let $\boldsymbol{\chi}^{(r)} = \{\chi_{b,t}^{(r)}, \chi_{s,t}^{(r)}, \chi_{f,t}^{(r)}\}_{t=1,\dots,T}$, then the following relation holds:

$$\boldsymbol{\chi}^{(r)} = M_T\left(\sqrt{c\Phi\gamma\xi}, \sqrt{\eta h}\right)\boldsymbol{\chi}^{(r-1)}, \tag{C.6}$$

where $M_T(\cdot,\cdot)$ is the matrix appearing in Claim 9.1. Following the argument we just detailed for $T \to \infty$, we then get that the largest eigenvalue of the *informative family* is asymptotically close to $\lambda_{\text{info},1} = \lambda_1^{\downarrow|\cdot|}\left(M_T\left(\sqrt{c\Phi\lambda\xi}, \sqrt{\eta h}\right)\right)$.

This analysis also allows us to describe the subsequent eigenvalues $\lambda_{\text{info},i\geq 2}$ belonging to the *informative family* that have a smaller modulus. These modes are metastable configurations of the branching process as in configuration 4 of Figure 9.5. In these modes, nodes belonging to different communities are still distinguished (hence the reason why these modes are *informative*), but the class identification σ_x may be reversed across time. This results in a state in which neighbours are more likely to change label than to keep it, hence they have *negative* label correlation and lead to negative values of χ. This means to relax the constraint $\chi > 0$ and thus no longer looking for the leading eigenvalue of M_T. From this intuition we argue that the subsequent informative eigenvalues of $B_{\xi,h}$ coincide with the subsequent eigenvalues of $M_T\left(\sqrt{c\Phi\gamma\xi}, \sqrt{\eta h}\right)$ which, as we already commented in the main text, are not necessarily real.

We now proceed extending our arguments to the *uninformative family* of isolated eigenvalues of $B_{\xi,h}$.

C.2.2 THE POSITION OF THE UNINFORMATIVE ISOLATED EIGENVALUES

As in the static case, not all stable configurations of the branching process of Appendix C.1 are informative. In particular, two nodes of $\mathcal{G}$ might be considered to belong to the same community only because they live at the

same time. Based on the technique detailed in Section C.2.1, we now describe the position of the eigenvalues forming the *uninformative family*. Although these eigenvalues are not informative, the awareness of their presence is crucial if one has to avoid to mistakenly use one of these for community reconstruction.

We proceed again by studying the largest of these eigenvalues (which is also the largest eigenvalue of $B_{\xi,h}$), to then extended the result to all the others. Let us denote $\{\lambda_{\text{uninfo},i}\}_{i=1,\dots,T}$ this second set of (trivial and non-informative) eigenvalues. The approximate Perron-Frobenius eigenvector $\boldsymbol{b} \in \mathbb{R}^{|\mathcal{E}_{\text{d}}|}$ can be written as

$$b_{ij}^{(r)} = \frac{1}{\lambda_{\text{uninfo},1}^{r}} \sum_{(wx)\,:\,\substack{d(jk,wx)=r \\ k\neq i}} W_{(jk)\to(wx)}. \tag{C.7}$$

According to this expression, we set $\sigma_x = 1$ for all nodes and thus the correlation between σ_x and σ_y is always unitary. Following the argument developed to determine the value of $\lambda_{\text{info},1}$, we then obtain

$$\lambda_{\text{uninfo},1} = \lambda_1^{\downarrow|\cdot|}\left(M_T\left(\sqrt{c\Phi\bar{\xi}}, \sqrt{h}\right)\right). \tag{C.8}$$

As in Section C.2.1, this eigenvalue is necessarily real and the subsequent eigenvalues of the *uninformative family* are given by the subsequent eigenvalues of $M_T\left(\sqrt{c\Phi\bar{\xi}}, \sqrt{h}\right)$ and can be complex. We underline once again that the ordering of $\{\lambda_{\text{info},i}\}_{i\geq 1}$ and $\{\lambda_{\text{uninfo},i}\}_{i\geq 1}$ is not known a priori.

So far we determined the position of the isolated eigenvalues under the assumption that the expectation of the approximate eigenvectors are significant. In order to know when this analysis holds, we have to study the variance of the entries of the approximate eigenvectors and see under what conditions it vanishes. This analysis will also allow us to determine the value of the radius of the bulk of $B_{\xi,h}$.

C.2.3 THE BULK EIGENVALUES OF $B_{\xi,h}$

To begin with, we investigate under which conditions the approximate eigenvector Equations (C.5, C.7) hold. We then proceed with a study of the variance of $g_{ij}^{(r)}$ (and $b_{ij}^{(r)}$). When the variance vanishes, the eigenvector is well approximated by its expectation and we conjecture it is isolated. On the contrary, when the variance diverges it is because it gets asymptotically close to the bulk of uninformative eigenvalues and is no longer isolated.

Let us first consider the eigenvector attached to $\lambda_{\text{info},1}$:

$$\mathbb{E}\left[\left(g_{ij}^{(r)}\right)^2\right] = \tag{C.9}$$

$$\frac{1}{\lambda_{\text{info},1}^{2r}} \sum_{(wx)\,:\,\substack{d(jk,wx)=r \\ k\neq i}} \left(W_{(jk)\to(wx)}^2 + \sum_{(vy)\,:\,\substack{d(jk,vy)=r \\ (vy)\neq(wx),k\neq i}} \sigma_x\sigma_y W_{(jk)\to(wx)} W_{(jk)\to(vy)} \right).$$

The first addend of (C.9) can be evaluated as done previously:

$$\mathbb{E}\left[\frac{1}{\lambda_{\text{info},1}^{2r}}\sum_{(wx)\,:\,\substack{d(jl,wx)=r\\ l\neq i}} W_{(jl)\to(wx)}^2\right] = O\left(\frac{\lambda_1^{\downarrow|\cdot|}\left(M_T(\sqrt{c\Phi\xi^2},h)\right)}{\lambda_{\text{info},1}^2}\right)^r.$$

If $\lambda_{\text{info},1}^2 < \lambda_1^{\downarrow|\cdot|}\left(M_T(\sqrt{c\Phi\xi^2},h)\right)$, this addend of diverges, and so does the variance of $g_{ij}^{(r)}$: in this case, $\boldsymbol{g}^{(r)}$ cannot be an approximate eigenvector of $B_{\xi,h}$. Consider next the second addend of Equation (C.9):

$$\begin{aligned}
&\mathbb{E}\left[\frac{1}{\lambda_{\text{info},1}^{2r}}\sum_{(wx)\,:\,\substack{d(jk,wx)=r\\ l\neq i}}\sum_{(vy)\,:\,\substack{d(jk,vy)=r\\ (vy)\neq(wx),k\neq i}} \sigma_x\sigma_y W_{(jk)\to(wx)}W_{(jk)\to(vy)}\right]\\
&= \frac{1}{\lambda_{\text{info},1}}\sum_{(wx)\,:\,\substack{d(jk,wx)=r\\ l\neq i}}\sum_{(vy)\,:\,\substack{d(jk,vy)=r\\ (vy)\neq(wx),k\neq i}} \mathbb{E}\left[\sigma_j\sigma_x W_{(jk)\to(wx)}\cdot\sigma_j\sigma_y W_{(jk)\to(vy)}\right]\\
&\approx \frac{1}{\lambda_{\text{info},1}}\sum_{(wx)\,:\,\substack{d(jk,wx)=r\\ l\neq i}}\mathbb{E}\left[\sigma_j\sigma_x W_{(jk)\to(wx)}\right]\sum_{(vy)\,:\,\substack{d(jk,vy)=r\\ (vy)\neq(wx),k\neq i}}\mathbb{E}\left[\sigma_j\sigma_y W_{(jk)\to(vy)}\right]\\
&\approx \frac{1}{\lambda_{\text{info},1}}\sum_{(wx)\,:\,\substack{d(jk,wx)=r\\ l\neq i}}\mathbb{E}\left[\sigma_j\sigma_x W_{(jk)\to(wx)}\right]\sum_{(vy)\,:\,\substack{d(jk,vy)=r\\ k\neq i}}\mathbb{E}\left[\sigma_j\sigma_y W_{(jk)\to(vy)}\right]\\
&= \mathbb{E}^2\left[g_{ij}^{(r)}\right],
\end{aligned}$$

where we exploited the fact that the paths $(jk\to wl)$ and $(jk\to vl)$ are asymptotically independent and that the number of paths leading to nodes a distance r from (jk) is exponentially large in r, unlike the number of paths leading to (vy) from (jk). We thus obtain that the variance $\mathbb{V}[g_{ij}^{(r)}]$ of $g_{ij}^{(r)}$ grows as

$$\mathbb{V}\left[g_{ij}^{(r)}\right] = O\left(\frac{\lambda_1^{\downarrow|\cdot|}\left(M_T(\sqrt{c\Phi\xi^2},h)\right)}{\lambda_{\text{info},1}^2}\right)^r. \tag{C.10}$$

As a consequence, the variance of $g_{ij}^{(r)}$ vanishes if and only if

$$\lambda_{\text{info},1} > \sqrt{\lambda_1^{\downarrow|\cdot|}\left(M_T(\sqrt{c\Phi\xi^2},h)\right)}.$$

Considering now the problem of evaluating the variance for all the $\{\lambda_{\text{info},i}\}_{i\geq 1}$ and $\{\lambda_{\text{uninfo},i}\}_{i\geq 1}$, note that, the variance is only determined by the first addend of Equation (C.9). This term does not depend on the configuration σ and is, therefore, the same for *all* the isolated eigenvectors. Consequently, for all the isolated eigenvectors, the variance vanishes if the corresponding eigenvalue is greater than $L_{\xi,h} = \sqrt{\lambda_1^{\downarrow|\cdot|}\left(M_T(\sqrt{c\Phi\xi^2},h)\right)}$, which is

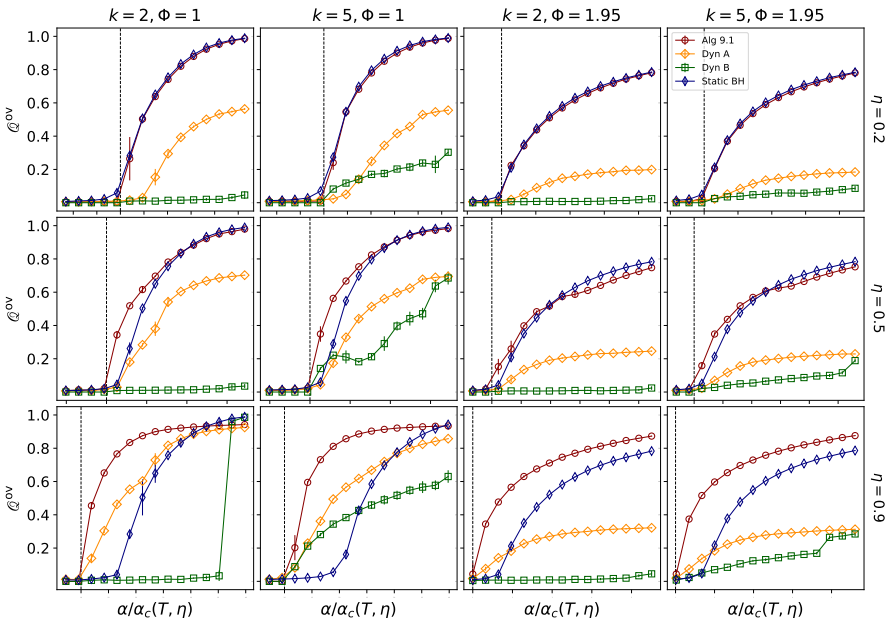


Figure C.1: Overlap comparison of Algorithm 9.1, the dynamic adjacency matrix of [KV20] (Dyn A), the dynamic non-backtracking of [Gha+16] (Dyn B) and the static Bethe-Hessian of [DCT19] (Static BH). The title of each row ant column indicates the values of η, k, Φ considered. For $\Phi \neq 1$ a power law degree distribution is adopted. The vertical line indicates the position of $\alpha/\alpha_c(T, \eta) = 1$. For all simulations: $c = 6$, $n = 25\,000$, $T = 4$. Averages are taken over 10 samples.

precisely the radius of the bulk of $B_{\xi,h}$, since an informative eigenvalue-eigenvector pair $(\lambda_{\text{info},i}, \boldsymbol{g}_i)$, (resp. $(\lambda_{\text{uninfo},i}, \boldsymbol{b}_i)$), for $B_{\xi,h}$ can only exist provided that $\lambda_{\text{info},i}$ (resp. $\lambda_{\text{uninfo},i}$) is greater than $L_{\xi,h}$, hence concluding the justification of Claim 9.2.

C.3 PERFORMANCE COMPARISON

This section compares numerically the performance of Algorithm 9.1 against the main spectral methods commented along Chapter 9. In Figure C.1 the algorithms are tested for a different number of classes, value of η and degree distribution. For $k > 2$ a symmetric setting with classes of equal size and $C_{ab} = c_{\text{out}}$ for all $a \neq b$ is considered, so that the spectral algorithm of [Gha+16] is still well defined. Figure C.1 indeed confirms that Algorithm 9.1 (i) benefits from high label persistence η; (ii) systematically outperforms the two considered competing dynamical sparse spectral algorithms [KV20], [Gha+16]; (iii) is capable of handling an arbitrary degree distribution.

To compare the performance of Algorithm 9.1 and the static Bethe-Hessian of [DCT19], the case of small and large values of α should be treated separately. Close to the transition, Algorithm 9.1 improves over the static Bethe-Hessian and this gets more evident as η increases: the joint solution of the

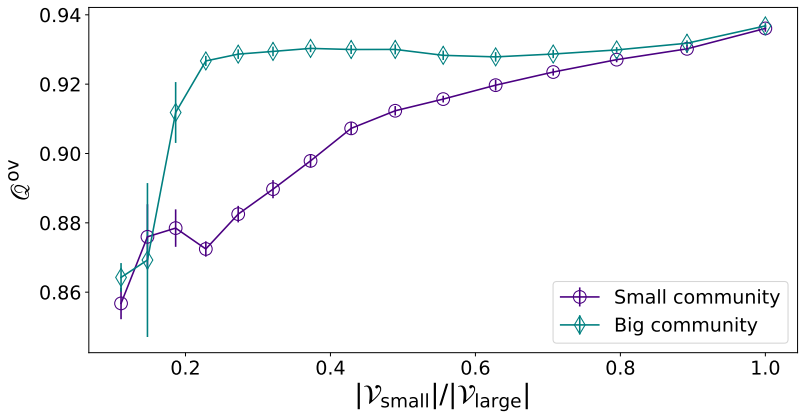


Figure C.2: Overlap averaged over time achieved by Algorithm 9.1 on graphs with two communities of different size, as a function of the ratio of the size of the two communities. For this simulation $n = 10\,000$, $T = 5$, $c = 6$, $\boldsymbol{\theta} = \mathbf{1}_n$, $\eta = 0.7$. The second largest eigenvalue of $C\Pi$ is fixed to $\lambda_2^{\downarrow}(C\Pi) = 4$. Averages over 15 samples.

problem at all times allows one to improve the clustering performance in the hard detection regime. For large values of α, instead, there seems to exist $\alpha^*(\eta)$ beyond which regularity only marginally improves the detection performance and Algorithm 9.1 performs equally (or slightly worse) than the static algorithm of [DCT19]. Here, Algorithm 9.1 suffers the sub-optimal choices commented in Section 9.3 made to obtain a practical algorithm achieving non-trivial reconstruction when close to $\alpha_c(T, \eta)$. On the opposite, the static Bethe-Hessian of [DCT19] is explicitly designed to optimally perform community detection for all values of α and any degree distribution, thereby justifying the two curves for large values of α.

A last remark concerns the capability of Algorithm 9.1 to recover communities of unequal sizes. Figure C.2 shows the accuracy of reconstruction of two communities of different size, as a function of the size of the smallest cluster over the size of the biggest. In order to obtain comparable results for different values of the ratio of the sizes of the two clusters, the following strategy is adopted: let $\Pi \in \mathbb{R}^{2\times 2}$ be the diagonal matrix defined so that Π_{ii} is the fraction of nodes belonging to class i ($\mathrm{Tr}(\Pi) = 1$), $C\Pi\mathbf{1}_2 = c\mathbf{1}_n$. For a given ratio Π_{11}/Π_{22}, the matrix C is constructed so to let the leading eigenvalue of $C\Pi$ equal to c, and the second eigenvalue (which controls the hardness of CD) equal to a fixed value. The overlap (averaged over time) is then evaluated independently over the large and small class, to keep this measure meaningful: in the case $|\mathcal{V}_{small}| \gg |\mathcal{V}_{large}|$, assigning all nodes to the same cluster would output a large overlap.