論文 / 著書情報
Article / Book Information

題目(和文)	スパース行列に対する行列分解の統計力学的解析
Title(English)	Statistical mechanics analysis of low-rank matrix factorization for sparse matrices
著者(和文)	野口千尋
Author(English)	Chihiro Noguchi
出典(和文)	学位:博士(理学), 学位授与機関:東京工業大学, 報告番号:甲第11671号, 授与年月日:2020年12月31日, 学位の種別:課程博士, 審査員:渡邊 澄夫,金森 敬文,高安 美佐子,三好 直人,山下 真,樺島 祥介
Citation(English)	Degree:Doctor (Science), Conferring organization: Tokyo Institute of Technology, Report number:甲第11671号, Conferred date:2020/12/31, Degree Type:Course doctor, Examiner:,,,,,
学位種別(和文)	博士論文
Type(English)	Doctoral Thesis

Doctoral Dissertation

Statistical mechanics analysis of low-rank matrix factorization for sparse matrices

Chihiro Noguchi

Department of Mathematical and Computing Science,
School of Computing,
Tokyo Institute of Technology

Supervisor: Professor Sumio Watanabe

Co-supervisor: Professor Yoshiyuki Kabashima

September 29, 2020

Abstract

In order to extract principal information from observed data, we often exploit the low-rank structure of the data. Low-rank matrix factorization is a popular method to find such structures and has a wide application range. This study addresses matrix completion and community detection as its two major applications, employing methods of statistical mechanics.

For matrix completion, we develop two algorithms, which are inspired from the cavity method. They are performable wit low computational costs and applicable to large scale data in parallel and distributed manners. For community detection, on the other hand, we argue the problem that the underlying ambiguity in real-world networks affects the performance in detecting community structures. More precisely, we analyze how the overlapping structures deteriorate the performance of spectral clustering, which is a popular algorithm for detecting communities, by using the replica method.

Contents

Chapter 1

Introduction

1.1 Background

An enormous amount of data is being generated everyday from various contexts. However, extracting useful information from them is a difficult task. To tackle this difficulty, fields such as machine learning and high-dimensional statistics have been receiving attention. Dimensionality reduction [1] plays a central role particularly when the observed data is significantly large and high-dimensional. This technique enables us not only to compress the data size but also to obtain principal features of the data. Due to its usefulness, it is often used as a fundamental technology in a wide range of domains and methods such as multivariate analysis, visualization [2], and feature extraction [3, 4].

Given N data points of M-dimensionality $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, these are often expressed by an N-by-M matrix X, whose ith row is $\boldsymbol{x}_i$ $(i = 1, \ldots, N)$. When the matrix sizes N and M are significantly large, a goal of the dimensionality reduction is to obtain a pair of compressed matrices $U \in \mathbb{R}^{N\times R}$ and $V \in \mathbb{R}^{M\times R}$, where $R < N, M$ so that X is approximated well by UV^T. Such methods are generically termed matrix factorization, which constitutes a core of various sophisticated dimensionality reduction methods that have been developed in recent years. There are many variants of matrix factorization depending on extra constraints imposed on the matrices. For instance, singular value decomposition is considered an essential example, for which the orthogonality is imposed among rows in both U and V. By this, we are able to interpret the factorized matrices as independent features of the observed data. Setting rank R much smaller than the original matrix size leads to another usage of matrix factorization. This assumption is reasonable in many practical settings, because the amounts of principal information contained in the large observed data can be much smaller than the data size. In fact, an example shown in Fig. 1.1 supports this assumption. In this example, the observed matrix is given by the image in Fig. 1.1a, and Fig. 1.1b shows how well the image is approximated by matrix factorization varying rank R. We can see that the necessary rank to construct the image sufficiently is much smaller than the image size $(= 256)$, because the MSE is suddenly reduced around $R = 30$. This indicates that the factorization into low rank matrices can be used as efficient lossy data compression techniques.

Low rank matrix factorization of another kind is used for matrices of the form of $Y = XX^T$, whose (i, j)th element y_{ij} represents the similarity between two data points $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$.

Data that represents relationships between data points is called the relational data, and it is usually expressed by a matrix like Y. Here, let us assume that Y is a sparse matrix. This indicates a data point is closely related to only few other data points. As before, one can extract principal information by employing dimensionality reduction for Y. Along this line, two applications, matrix completion and community detection, are very popular and have high demands for practical use.

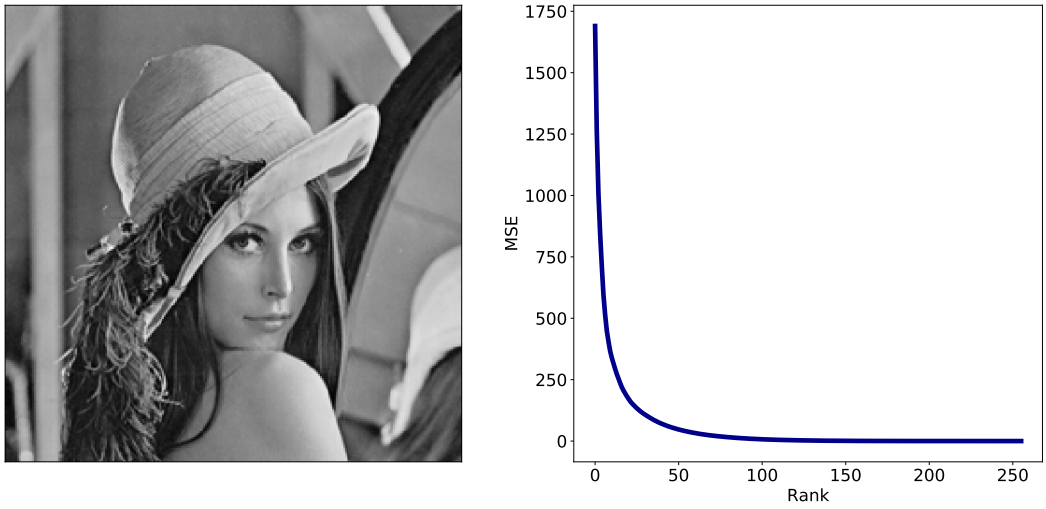
1.2 Purpose

In this thesis, we address issues of low-rank matrix factorization utilizing notions and knowledge of statistical mechanics. One direction is development of practically efficient algorithms for matrix factorization. In real-world problems, data are often given as considerably large matrices, and thus the practical algorithms are required to be performable with low computational costs in parallel and/or distributed manners. For answering such demands, we develop approximate algorithms for matrix factorization, which we call cavity-based matrix factorization (CBMF) and approximate cavity-based matrix factorization (ACBMF), borrowing an idea from the cavity method. Their usefulness is tested by applications to matrix completion problems.

The other direction is theoretical analysis of possibilities and limitations of matrix factorization. For this, we take up a community detection problem, and examine how complex structures that could be contained in real-world data affects the performance of finding communities. Spectral clustering [5], which is based on the low-rank matrix factorization, is a popular method for community detection. Although this algorithm exhibits excellent performance when clusters are clearly separated in given networks, complex structures underlying in real-world data often deteriorate the detection performance. It is, therefore, of importance to clarify the mechanism of the deterioration. To this end, we focus particularly on overlapping structure of communities, and analyze the performance of spectral clustering employing the replica method to random graph models. Our analysis clarifies how the structural information is lost from the leading eigenvector depending on the model parameters related to the size and density of the overlapping structure.

1.3 Organization

The rest of the paper is organized as follows. In Chapter 2, we summarize notations and knowledge necessary for subsequent chapters. In Chapter 3, we develop two novel matrix completion algorithms, which we call cavity-based matrix factorization (CBMF) and approximate cavity-based matrix factorization (ACBMF). In Chapter 4, we investigate the deterioration mechanism of spectral clustering for overlapping structures by using the replica method. Finally, Chapter 5 presents a summary of the achievements and possible future work.



(a) (b)

Figure 1.1: (a) Observed matrix. (b) Mean squared error (MSE) of the approximated matrix as a function of the rank of the approximated matrix.

Chapter 2

Preliminaries

In this chapter, we summarize notions and knowledge necessary for subsequent chapters. In Sec. 2.1, we introduce low-rank matrix factorization. In Secs. 2.2 and 2.3, as its two popular applications, community detection and matrix completion, are introduced, respectively.

2.1 Low-rank matrix factorization

Matrix factorization has a number of variants depending on the purpose and the imposed constraints. The current study particularly concentrates on the case where a matrix is factorized into a pair of two matrices of smaller sizes. This is called the low-rank matrix factorization. In the following, we introduce the singular value decomposition. It is regarded as the simplest example of problems addressed in this paper.

2.1.1 Singular value decomposition (SVD)

Singular value decomposition (SVD) is a common technique in linear algebra, and is applied in a wide range of fields. SVD can be considered as the generalization of an eigenvalue decomposition. The eigenvalue decomposition is defined only for a squared matrix, whereas SVD can also be applied to a non-square matrix, and both are identical for a symmetry matrix. For matrix $Y \in \mathbb{R}^{N\times M}$ $(N \leq M)$, we can find the following factorization:

$$Y = U_0 \Sigma_0 V_0^T. \tag{2.1}$$

Here, $U_0 \in \mathbb{R}^{N\times N}$ and $V_0 \in \mathbb{R}^{M\times N}$ are the orthogonal matrices, i.e, $U_0 U_0^T = U_0^T U_0 = I_N$ and $V_0^T V_0 = I_N$. $I_N \in \mathbb{R}^{N\times N}$ is the identity matrix. $\Sigma_0 \in \mathbb{R}^{N\times N}$ is the diagonal matrix whose diagonal elements are the singular values $\sigma_1, \ldots, \sigma_N$, and we can assume $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_N$ without loss of generality. Furthermore, we describe the column vectors of U_0 and V_0 as $\boldsymbol{u}_i \in \mathbb{R}^N$ $(i = 1, \ldots, N)$ and $\boldsymbol{v}_i \in \mathbb{R}^M$ $(i = 1, \ldots, N)$, respectively. The column vectors $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are referred to as the *left singular vectors* and *right singular vectors*. Here, we can easily confirm that the squared eigenvalues of $Y^T Y$ are identical to the singular values of Y

because $Y^TY = V_0\Sigma_0^2V_0^T$. Moreover, SVD in Eq. (2.1) can be recast as

$$Y = \sum_{i=1}^{N} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T. \tag{2.2}$$

This form is useful to discuss a low-rank approximation later.

Low-rank approximation

Given rank-R matrix $Y_R \in \mathbb{R}^{N\times M}$, its singular values can be $\tilde{\sigma}_1 \geq \cdots \geq \tilde{\sigma}_R > 0$ and $\tilde{\sigma}_{R+1} = \cdots = \tilde{\sigma}_N = 0$. In this case, using Eq. (2.2), SVD of Y_R can be expressed as

$$Y_R = \sum_{i=1}^{R} \tilde{\sigma}_i \tilde{\boldsymbol{u}}_i \tilde{\boldsymbol{v}}_i^T, \tag{2.3}$$

where $\tilde{\boldsymbol{u}}_i$ and $\tilde{\boldsymbol{v}}_i$ $(i = 1, \ldots, R)$ are the left and right singular vectors of Y_R, respectively. From Eq. (2.3), a rank-R matrix can be interpreted as a weighted sum of rank-1 matrices that consist of left and right singular vectors, and the weights are given by the singular values. Therefore, we can consider an approximation that excludes the singular values with small magnitude. This approximation is called the low-rank approximation. The low-rank approximation of Y (the rank of Y is greater than R) is formulated as

$$Y \approx \sum_{i=1}^{R} \sigma_i \boldsymbol{u}_i \boldsymbol{v}_i^T. \tag{2.4}$$

This approximation sets the smaller singular values than σ_R to 0. As a result, the approximated matrix becomes a rank-R matrix. This approximation is called the rank-R low-rank approximation.

We can show that the low-rank approximation is optimal in terms of minimizing the squared error. In other words, the solution of the minimization problem

$$\hat{X} = \underset{\mathrm{rank(X)=R}}{\operatorname{argmin}} \|Y - X\|_F^2 \tag{2.5}$$

is given by the rank-R low-rank approximation of Y. Here, $\hat{X} = U_R\Sigma V_R^T$, where the column vectors of U_R and V_R are formed by the top R left and right singular vectors of Y, respectively. In the following, we verify Eq. (2.5) in brief. A strict proof is available in Refs. [6, 7]. The minimization problem in Eq. (2.5) can be rewritten as

$$\min_{\substack{Q\in\mathbb{R}^{N\times R} \\ Q^TQ=I_R}} \min_{B\in\mathbb{R}^{M\times R}} \|Y - QB^T\|_F^2. \tag{2.6}$$

Here, X can be generally factorized into $Q \in \mathbb{R}^{N\times R}$ and $B \in \mathbb{R}^{M\times R}$, where Q is an orthogonal

matrix. Then, solving for B, we obtain

$$\min_{\substack{Q\in\mathbb{R}^{N\times R}\\ Q^TQ=I_R}} \|Y - QQ^TY\|_F^2 \tag{2.7}$$

$$= \min_{\substack{Q\in\mathbb{R}^{N\times R}\\ Q^TQ=I_R}} \mathrm{Tr}\left((Y - QQ^TY)^T(Y - QQ^TY)\right) \tag{2.8}$$

$$= \max_{\substack{Q\in\mathbb{R}^{N\times R}\\ Q^TQ=I_R}} \mathrm{Tr}(Q^TYY^YQ). \tag{2.9}$$

To derive the last equation, a constant term was ignored. Introducing a Lagrange multiplier $\Lambda \in \mathbb{R}^{R\times R}$, we obtain the Lagrange function as

$$g(Q) = \mathrm{Tr}(Q^TYY^YQ) - \mathrm{Tr}\left(\Lambda^T(Q^TQ - I_R)\right), \tag{2.10}$$

where Λ is a diagonal matrix $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_R)$. Therefore, the optimal solution is given by solving

$$SQ = Q\Lambda, \tag{2.11}$$

where we put $S = YY^T$. Letting $Q = (\boldsymbol{q}_1, \ldots, \boldsymbol{q}_R)$, Eq. (2.11) can be described as

$$(S\boldsymbol{q}_1, \ldots, S\boldsymbol{q}_R) = (\lambda_1\boldsymbol{q}_1, \ldots, \lambda_R\boldsymbol{q}_R). \tag{2.12}$$

Therefore, each column of Eq.(2.12) is regarded as a different eigenvalue problem of S. The optimal solution of Eq. (2.9) is given by the sum of the eigenvalues, the columns of Q are the top R eigenvectors of S. In the light of $B = Q^TY$, $\hat{X}$ is given by the rank-R SVD of Y.

The low-rank approximation allows us to compress the amount of data. The original matrix Y has NM entries, whereas the factorized matrices have $(N + M + R)R$ entries. When $R \ll N, M$, the amount of data is significantly compressed. In addition, the necessary computational cost is also reduced compared to the full-rank SVD. The computational cost of the full-rank SVD is $O(N^3)$ in general, whereas the rank-R SVD can be performed with computational cost of $O(N^2)$. Here, when Y is a sparse matrix, its computational cost can be further reduced. Lanczos algorithm [8] is an efficient algorithm for the rank-R SVD. The Lanczos algorithm is based on the powered method and computes the top R singular values by repeating matrix-vector multiplications. The matrix-vector multiplication for a sparse matrix can be computed with computational cost of $O(N)$ because its computation for zero entries can be skipped.

SVD for clustering

In this section, we discuss SVD from the perspective of the clustering. This perspective is of value when considering spectral clustering in Sec.2.2.1. Here, we regard Y as an observed matrix and its each row $\boldsymbol{y}_i \in \mathbb{R}^M$ $(i = 1 \ldots, N)$ as the ith observed data point. Then, each row of SVD in Eq. (2.1) can be recast as

$$\boldsymbol{y}_i = \sum_{j=1}^{N} u_{ij}\sigma_j\boldsymbol{v}_j^T, \tag{2.13}$$

where u_{ij} is the (i,j)th element of U. In the context of principal component analysis (PCA), the right singular vector $\boldsymbol{v}_j$ is referred to as the jth principal component, the singular value σ_j is interpreted as the importance of $\boldsymbol{v}_j$, and u_{ij} can be understood as the weight of $\boldsymbol{v}_j$ for observed data $\boldsymbol{y}_i$. From the definition of SVD, the principal components are orthogonal each other. Namely, the principal components represent the independent features of the observed data.

Let us consider the rank-R low-rank approximation for Eq. (2.13). Then, the observed data can be approximated as

$$\boldsymbol{y}_i \approx \sum_{j=1}^{R} u_{ij}\sigma_j \boldsymbol{v}_j^T. \tag{2.14}$$

This approximation indicates that we pick up top R important features and approximate the observed data by using only them. From the perspective of the clustering, these features correspond to the clusters, and u_{ij} represents the strength that observed data $\boldsymbol{y}_i$ belongs to the jth cluster.

However, the above mentioned interpretation has a practical issue that weight u_{ij} can take a negative value. When we consider the general setting of the clustering, each data point is allowed to belong to several clusters simultaneously (overlapping clusters). In this case, we hope that a clustering algorithm outputs a probability or an assignment of belonging to each cluster. However, when weight u_{ij} takes a negative value, we cannot provide such interpretation for the factorized matrices. Therefore, $\boldsymbol{u}_i$ is often used as a low dimensional feature of $\boldsymbol{y}_i$. The features can be used for inputs of another clustering algorithm such as k-means clustering. Although these procedures seem to be redundant at a glance, it is essential when the observed matrix represents relational data. This type of clustering is called the spectral clustering (see Sec. 2.2.1 in detail).

Note that to perform the clustering directly by using the low-rank matrix factorization, we often impose the non-negative constraints on the factorized matrices. This is called the non-negative matrix factorization (NMF) [9].

2.2 Community detection

Community detection is the task to find a specific structure in a graph. The graph here represents a basic data structure that consists of nodes and edges. A node represents a data point, and an edge represents a relation between a pair of nodes. The goal of the community detection is to identify communities whose nodes are more densely connected in a graph. Here, we denote an undirected graph as $G = (V, E)$, where V ($|V| = N$) is a set of nodes and E ($|E| = m$) is a set of edges. The graph is represented by $N \times N$ adjacency matrix A, where $A_{ij} = 1$ when a pair of nodes i and j is connected by an edge and $A_{ij} = 0$ otherwise. In this paper, the graph is assumed to be sparse, namely the number of neighbors of a node is much smaller than the system size. The community detection is a popular task and a number of algorithms for it have been proposed. This paper focuses on the spectral clustering particularly as a method based on the low-rank matrix factorization.

2.2.1 Spectral clustering

Spectral clustering [5] is a popular method for community detection. It is based on the low-rank matrix factorization of a matrix associated with an adjacency matrix of a graph. Such a matrix is called the Laplacian matrix. In the simplest case, the adjacency matrix itself is used as the Laplacian matrix. However, its performance is known to be poor in many cases. Therefore, we usually use more sophisticated Laplacian matrices. In fact, they are closely related to the objective function of the other community detection methods. In this subsection, we introduce several objective functions in different contexts of the community detection, including modularity maximization, graph partitioning and statistical inference. Then, we show that they can be reduced to a variant of the spectral clustering by the continuous relaxation. These discussions are also available in Refs. [10, 11].

Modularity maximization

Modularity is the measure of the strength of community structures in a graph [12]. We can show that maximizing the modularity leads to the spectral clustering with a modularity matrix.

Given a group assignment, the definition of modularity is given by the difference between the fraction of edges whose both endpoints belong to the same group and the expected fraction when edges are distributed at random. A pair of nodes in the same group is expected to be connected with a higher probability than that in the uniform random graph. Therefore, by finding a group assignment so that modularity is maximized, the assignment is expected to reflect on the community structures in the graph. When the number of groups is two, the definition of modularity $\mathcal{Q}_2$ is formulated as

$$\mathcal{Q}_2 = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \frac{s_i s_j + 1}{2}. \tag{2.15}$$

Here, $\boldsymbol{s} = [s_i]$ $(i = 1, \ldots, N)$, $s_i \in \{-1, 1\}$ is the group assignment, and m $(= \sum_{ij} A_{ij}/2)$ is the number of edges in the graph. The first term is the adjacency matrix that represents the graph structure, and the second term is the probability that a pair of nodes (i, j) connects randomly when their degrees are given. Due to the factor $(s_i s_j + 1)/2$, the objective function takes a non-zero value only when the pair of nodes belongs to the same group. Equation (2.15) can be recast as

$$\mathcal{Q}_2 = \frac{1}{4m} \boldsymbol{s}^T M \boldsymbol{s}, \tag{2.16}$$

where

$$M_{ij} = A_{ij} - \frac{d_i d_j}{2m} \tag{2.17}$$

is the (i, j)th element of modularity matrix M. Note that the last term in the last factor in Eq. (2.15) can be ignored because of $\sum_{ij} d_i d_j / 2m = \sum_{ij} A_{ij}$.

Maximizing Eq. (2.16) with respect to $\boldsymbol{s}$ is computationally difficult because we require calculating for exponential patterns (2^N). Therefore, a heuristic approach is needed to solve the problem in practice. A standard heuristics here is a continuous relaxation; the elements

of $\boldsymbol{s}$ are relaxed to taking any real values. As a result, the modularity maximization problem with the continuous relaxation is obtained as

$$\begin{aligned} \max_{\boldsymbol{x}} \quad & \frac{1}{N}\boldsymbol{x}^T M \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{x}^T \boldsymbol{x} = N. \end{aligned} \tag{2.18}$$

Here, elements of $\boldsymbol{x}$ can take real values, which were introduced instead of $\boldsymbol{s}$. Its constraint is required to prevent the optimal solution from diverging. By introducing a Lagrange multiplier, we can confirm that the solution of this problem is given by the first eigenvector of M.

Expanding to the case with more than two groups is easy. When the number of groups is R, Eq. (2.15) can be expanded as

$$\mathcal{Q}_R = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{d_i d_j}{2m}\right)\sum_{r=1}^{R} s_{ir}s_{jr}. \tag{2.19}$$

Here, $S \in \mathbb{R}^{N\times R}$ is the assignment matrix whose (i,r)th element $s_{ir} = 1$ if node i belongs to group r and $s_{ir} = 0$ otherwise, and its orthogonality is assumed, i.e, $S^T S = I_R$. Then Eq. (2.19) can be recast as

$$\mathcal{Q}_{\mathcal{R}} = \frac{1}{2m}\operatorname{Tr}\left(S^T M S\right). \tag{2.20}$$

By performing the continuous relaxation of S, minimizing Eq. (2.20) with the orthogonality constraints becomes identical to Eq. (2.9). This indicates that the optimal solution is given by $\hat{X} \in \mathbb{R}^{N\times R}$ whose columns consist of the top R eigenvectors of M. As a result, finding R groups in a graph in terms of the modularity maximization reduces to the rank-R matrix factorization. Note that we are required to additionally perform a clustering algorithm, such as k-means clustering, to get a group assignment because $\hat{X}$ is no longer viewed as the assignment matrix.

Graph partitioning

Graph partitioning is also a popular approach for community detection. The goal of the graph partitioning is to cut a graph so that nodes in the cut subgraphs are connected more densely. This indicates that this goal is similar to that of the community detection, which enables us to use the graph partitioning techniques also for the community detection.

The simplest definition of the graph partitioning problem is given as the minimum cut problem. It aims to find a partition so that the sum of edge weights connected inter the cut subgraphs are minimized. When the number of partitions $R = 2$, it is formulated as minimizing the following objective function:

$$\operatorname{cut}(C, \overline{C}) = \sum_{i\in C, j\in\overline{C}} A_{ij}. \tag{2.21}$$

Here, $C \subset V$ is the subset of nodes in a cut subgraph and $\overline{C}$ is the complement of the subsect C. From this, when the number of partitions is given by R, the objective function of the

minimum cut problem is obtained as

$$\mathrm{cut}(C_1, \ldots, C_R) = \sum_{r=1}^{R} \mathrm{cut}(C_r, \overline{C}_r), \tag{2.22}$$

where C_r $(r = 1, \ldots, R)$ is the subset of nodes in the rth partitioned subgraph. When $R = 2$ and the elements of the adjacency matrix are non-negative, some algorithms that can solve the problem in polynomial time are known [13]. On the other hand, when $R \geq 3$, the problem is NP-hard and some heuristics are required in practice. Furthermore, it is known that we get only a trivial partition by solving the minimum cut problem in many practical cases. Namely, a partition that one node is isolated and the other nodes are grouped is often provided as the optimal partition of the minimum cut problem. To avoid this and get a more meaningful partition, giving a penalty to the size of each partition is an effective approach. Depending on the types of the imposed penalties, there are two major approaches: Raitio Cut (Rcut) [14] and Normalized Cut (Ncut) [15]. The objective functions of these graph partitioning are defined as follows:

$$\mathrm{Rcut}(C_1, \ldots, C_R) = \sum_{r=1}^{R} \frac{\mathrm{cut}(C_r, \overline{C}_r)}{|C_r|}, \tag{2.23}$$

$$\mathrm{Ncut}(C_1, \ldots, C_R) = \sum_{r=1}^{R} \frac{\mathrm{cut}(C_r, \overline{C}_r)}{\mathrm{vol}(C_r)}. \tag{2.24}$$

Here, $|C|$ is the number of nodes in the subset C, and $\mathrm{vol}(C)$ is the sum of edge weights connected to the nodes in the subset C. In both definitions, the larger the size of each partition ($|C_r|$ or $\mathrm{vol}(C_r)$) is, the smaller the objective function is. Thus, more balanced and meaningful partitions are expected to be found.

First, let us see Rcut in detail and show its objective function (2.23) can be reduced to a variant of the spectral clustering. We define the normalized group assignment $\boldsymbol{s}_r \in \mathbb{R}^N$ $(r = 1, \ldots, R)$, whose ith element s_{ir} is given by

$$s_{ir} = \begin{cases} \frac{1}{\sqrt{|C_r|}} & \text{if } \ i \in C_r \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.25}$$

Besides, we define a Laplacian matrix as

$$L = D - A, \tag{2.26}$$

which are called the unnormalized graph Laplacian. Here, D is the diagonal matrix, whose ith diagonal element is $d_i = \sum_{j=1}^{N} A_{ij}$. Now we focus on the following quadratic form.

$$\boldsymbol{s}_r^T L \boldsymbol{s}_r = \boldsymbol{s}_r^T (D - A) \boldsymbol{s}_r \tag{2.27}$$

$$= \sum_{ij} A_{ij} (s_{ir}^2 - s_{ir} s_{jr}) \tag{2.28}$$

$$= \frac{1}{2} \sum_{ij} A_{ij} (s_{ir} - s_{jr})^2. \tag{2.29}$$

To derive the third equation, we used the fact that A is a symmetric matrix. Inserting Eq. (2.25) into (2.29), we obtain

$$\boldsymbol{s}_r^T L \boldsymbol{s}_r = \frac{1}{2} \sum_{\substack{i \in C_r \\ j \in \overline{C}_r}} A_{ij} \frac{1}{|C_r|} + \frac{1}{2} \sum_{\substack{i \in \overline{C}_r \\ j \in C_r}} A_{ij} \frac{1}{|C_r|} \tag{2.30}$$

$$= \frac{\mathrm{cut}(C_r, \overline{C}_r)}{|C_r|}. \tag{2.31}$$

To derive the last equation, we used Eq. (2.21). Accordingly, the objective function of Rcut (2.23) can be rewritten as

$$\mathrm{Rcut}(C_1, \ldots, C_R) = \sum_{r=1}^{R} \boldsymbol{s}_r^T L \boldsymbol{s}_r = \mathrm{Tr}(S^T L S), \tag{2.32}$$

where $S \in \mathbb{R}^{N \times R}$ is the normalized assignment matrix, whose rth column vector is $\boldsymbol{s}_r$. Here, the orthogonal constraint is imposed on S. After all, Rcut objective function becomes

$$\begin{aligned} \min_{\{C_1, \ldots C_R\}} \quad & \mathrm{Tr}(S^T L S) \\ \text{subject to} \quad & S^T S = I_R. \end{aligned} \tag{2.33}$$

However, this problem is NP-hard. Thus, we again relax S to real matrix $X \in \mathbb{R}^{N \times R}$. Rcut objective function is then reduced to

$$\begin{aligned} \min_{X \in \mathbb{R}^{N \times R}} \quad & \mathrm{Tr}(X^T L X) \\ \text{subject to} \quad & X^T X = I_R. \end{aligned} \tag{2.34}$$

This optimization problem is analogical to Eq. (2.9). Its optimal solution is given by the smallest R eingenvectors of L.

Similarly to Rcut, we can show that the objective function of Ncut (2.24) is reduced to a variant of the spectral clustering. Because its derivation is similar to that of Rcut, we will show it in brief here. We define the normalized group assignment $\boldsymbol{t}_r \in \mathbb{R}^N$ $(r = 1, \ldots, R)$, whose ith element t_{ir} is given by

$$t_{ir} = \begin{cases} \frac{1}{\sqrt{\mathrm{vol}(C_r)}} & \text{if } \ i \in C_r \\ 0 & \text{otherwise}\,. \end{cases} \tag{2.35}$$

Similarly to Rcut, the quadratic form with respect to $\boldsymbol{t}$ and L is related to the minimum cut objective function, i.e.,

$$\boldsymbol{t}_r^T L \boldsymbol{t}_r = \frac{\mathrm{cut}(C_r, \overline{C}_r)}{\mathrm{vol}(C_r)}. \tag{2.36}$$

Therefore,

$$\mathrm{Ncut}(C_1, \ldots, C_R) = \mathrm{Tr}(T^T L T), \tag{2.37}$$

where $T \in \mathbb{R}^{N \times R}$ is the normalized assignment matrix whose rth column is $\boldsymbol{t}_r$. In additoin, we can confirm that there are orthogonal constraints with respect to T as follows.

$$\boldsymbol{t}_r^T D \boldsymbol{t}_r = \sum_i d_i t_{ir} t_{ir} = \sum_{i \in C_r} \frac{d_i}{\mathrm{vol}(C_r)} = 1, \tag{2.38}$$

$$\boldsymbol{t}_r^T D \boldsymbol{t}_s = \sum_i d_i t_{ir} t_{is} = 0. \quad (r \neq s) \tag{2.39}$$

Therefore,

$$T^T D T = I_R. \tag{2.40}$$

From Eqs. (2.37) and (2.40), we can recast the minimizing problem of Ncut as

$$\begin{aligned} \min_{\{C_1, \ldots C_R\}} \quad & \mathrm{Tr}(T^T L T) \\ \text{subject to} \quad & T^T D T = I_R. \end{aligned} \tag{2.41}$$

Similarly to Rcut, we relax T to real matrix $X \in \mathbb{R}^{N \times R}$. As a result, the continuous relaxed Ncut problem are obtained by

$$\begin{aligned} \min_{X \in \mathbb{R}^{N \times R}} \quad & \mathrm{Tr}(X^T L X) \\ \text{subject to} \quad & X^T D X = I_R. \end{aligned} \tag{2.42}$$

The optimal solution of this problem is given by the smallest R generalized eigenvectors. The generalized eigenvalue problem is formulated as $L\boldsymbol{x} = \lambda D\boldsymbol{x}$. Here, by replacing as $X' = D^{\frac{1}{2}} X$, Eq. (2.42) can be recast as

$$\begin{aligned} \min_{X' \in \mathbb{R}^{N \times R}} \quad & \mathrm{Tr}(X'^T \tilde{L} X') \\ \text{subject to} \quad & X'^T X' = I_R, \end{aligned} \tag{2.43}$$

where

$$\tilde{L} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \tag{2.44}$$

is a Laplacian matrix, which is called the normalized graph Laplacian. From Eq. (2.43), the optimal solution is again given by the smallest R eigenvectors of $\tilde{L}$.

Statistical inference

Statistical inference is also a widely used approach for community detection. In this approach, a generative model is assumed and the observed data are assumed to be sampled from the model. A goal of the statistical inference approach is to find the most fitted parameters of the model for the observed data. In the context of community detection, stochastic block model (SBM) is a popular generative model because it holds the simplicity, expressiveness and extensibility. The simplicity comes from the block structure of SBM. The probability a pair of nodes is connected by an edge is determined by the block (group) to which the nodes belong. Moreover, the expressiveness of SBM is high because it can generate a wide variety of

graphs not only with community structures but also with disassortative structures [16], core-periphery structures [17] and so on. In terms of the extensibility, a number of variants have been proposed such as labeled SBM [18], degree corrected SBM [19], overlapping SBM [20] and microcanonical SBM [21]. The detailed discussion of SBM is available in a comprehensive review [22]. In the following, we introduce the standard SBM and its inference method, and then we discuss the equivalence with the spectral clustering. The following discussion is also available in [10, 11, 23].

In the standard SBM, node i is assumed to belong to only one group denoted as t_i, and the probability that a pair of nodes (i, j) is connected by an edge is denoted as $\rho_{t_i t_j}$. Each edge is drawn independently and randomly with probability ρ_{rs}, which is the (r, s)th element of the $R \times R$ affinity matrix $\boldsymbol{\rho} = [\rho_{rs}]$, $0 \leq \rho_{rs} \leq 1$. The probability of being generated a graph instance is expressed as

$$P(A|R, \boldsymbol{t}, \boldsymbol{\rho}) = \prod_{i<j} \rho_{t_i t_j}^{A_{ij}} \left(1 - \rho_{t_i t_j}\right)^{1-A_{ij}}, \tag{2.45}$$

where R is the number of groups and $\boldsymbol{t}$ is the group assignment, whose ith element is t_i. To infer $\boldsymbol{t}$, we often adopt the Bayesian approach. Here, prior $P(\boldsymbol{t}|\boldsymbol{\gamma}) = \prod_i \gamma_{t_i}$, where γ_r is the fraction of nodes in the rth group, is introduced to consider partition function $P(A|\boldsymbol{\rho}, \boldsymbol{\gamma}) = \sum_{\boldsymbol{t}} P(A|R, \boldsymbol{t}, \boldsymbol{\rho}) P(\boldsymbol{t}|R, \boldsymbol{\gamma})$ of posterior $P(\boldsymbol{t}|A, R, \boldsymbol{\rho}, \boldsymbol{\gamma})$. However, inferring the posterior is computational intractable because of $\sum_{\boldsymbol{t}}$ operation, and thus we usually use an approximate inference such as variational Bayes [24], belief propagation [25] and MCMC [26]. As well as $\boldsymbol{t}$, the model contains parameters $\boldsymbol{\rho}$ and $\boldsymbol{\gamma}$, which must be inferred in general. To realize both inferences, EM algorithm can be used [24, 27].

In contrast to the Bayesian approach, in the following, we consider to find optimal $\boldsymbol{t}$ so that the likelihood function is maximized. We show that its objective function is equivalent to that of the modularity maximization when $\boldsymbol{\rho}$ is given and the graph is sparse. We now introduce an extension of the standard SBM. In Eq. (2.45), the probability a pair of nodes (i, j) is connected by an edge follows the Bernoulli distribution with mean ρ_{t_i, t_j}. Here, we replace the Bernoulli distribution with the Poisson distribution with mean ρ_{t_i, t_j}. As a result, we obtain

$$Q(A|R, \boldsymbol{t}, \boldsymbol{\rho}) = \prod_{i<j} \frac{\rho_{t_i t_j}^{A_{ij}}}{A_{ij}!} e^{-\rho_{t_i t_j}}. \tag{2.46}$$

This Poisson SBM is often used because of the theoretically tractable. Actually, the random variables that follow Bernoulli and Poisson SBMs behave roughly in the same way when ρ_{t_i, t_j} is sufficiently small [28]. Considering the log-likelihood of the two models, we obtain

$$\log P(A|R, \boldsymbol{t}, \boldsymbol{\rho}) \approx \sum_{i<j} \left(A_{ij} \log \rho_{t_i t_j} - \rho_{t_i t_j} + A_{ij} \rho_{t_i t_j}\right), \tag{2.47}$$

$$\log Q(A|R, \boldsymbol{t}, \boldsymbol{\rho}) \approx \sum_{i<j} \left(A_{ij} \log \rho_{t_i t_j} - \rho_{t_i t_j}\right). \tag{2.48}$$

Here, the $O(1/N)$ and constant terms are ignored. Eqs. (2.47) and (2.48) are equivalent except for the slight shift by the last term in Eq. (2.47). Note that although the Poisson SBM

has a possibility to generate multiple edges, its probability is much small when $\rho_{t_i,t_j} \ll 1$. Then, we consider an extension to the degree-corrected SBM [19]. Actually, it is known that the standard SBM rarely provides a good fit for real-world networks because of the tight assumption for the degree distribution. Although degrees of nodes in the same group follow the same Poisson distribution in the standard SBM, real-world networks usually have a heterogeneous non-Poisson degree distribution. To improve the fit for the real-world networks, an additional parameter θ_i is often introduced for each node. The modified generative model is given by

$$Q_D(A|R, \boldsymbol{t}, \boldsymbol{\rho}) = \prod_{i<j} \frac{\theta_i \theta_j \rho_{t_i t_j}^{A_{ij}}}{A_{ij}!} e^{-\theta_i \theta_j \rho_{t_i t_j}}. \tag{2.49}$$

Here, degree d_i is often used for the parameter θ_i. The log-likelihood of Eq. (2.49) is

$$\log Q_D(A|R, \boldsymbol{t}, \boldsymbol{\rho}) \approx \sum_{i<j} \left(A_{ij} \log \rho_{t_i t_j} - d_i d_j \rho_{t_i t_j} \right), \tag{2.50}$$

where again the $O(1/N)$ and constant terms are ignored. Here, we assume that the number of groups is two and the graph has an assortative structure, namely the diagonal elements of $\boldsymbol{\rho}$ are given by ρ_{in} and the others are ρ_{out} ($\rho_{\text{in}} > \rho_{\text{out}}$). Then, we express each element of $\boldsymbol{\rho}$ as follows.

$$\rho_{t_i t_j} = \frac{1}{2} \left((\rho_{\text{in}} + \rho_{\text{out}}) + \delta_{t_i t_j} (\rho_{\text{in}} - \rho_{\text{out}}) \right), \tag{2.51}$$

$$\log \rho_{t_i t_j} = \frac{1}{2} \left(\log \rho_{\text{in}} \rho_{\text{out}} + \delta_{t_i t_j} \log \frac{\rho_{\text{in}}}{\rho_{\text{out}}} \right), \tag{2.52}$$

where $\delta_{t_i t_j}$ represents the Kronecker's delta. Inserting Eqs. (2.51) and (2.52) into (2.50), we obtain

$$B \sum_{ij} \left(A_{ij} - \gamma d_i d_j \right) \delta_{t_i t_j} + C, \tag{2.53}$$

where $\gamma = (p_{\text{in}} - p_{\text{out}})/(\log p_{\text{in}} - \log p_{\text{out}})$ is the so-called resolution parameter [23]. B and C are constants depending only on ρ_{in} and ρ_{out}. Given $\boldsymbol{\rho}$, Eq. (2.53) is equivalent to the modularity objective function (2.15) if the resolution parameter is set appropriately. Therefore, this problem reduces to the spectral clustering with the modularity matrix.

2.3 Matrix completion

Recent technological advances triggered the generation and accumulation of significant amounts of data. In response to the trend, several methods are proposed to extract useful information from them. This produced significant results in various fields including science and engineering. A typical example can be found in collaborative filtering, which is a methodology that is used in recommender systems [29]. As a comprehensive example, we consider a user-movie matrix $Y \in \mathbb{R}^{N \times M}$, where N and M denote the number of users and movies, respectively, and an entry of Y, y_{ij}, denotes rating from user i movie j. Users normally evaluate only a small

fraction of movies, and thus most entries of Y are missing. Under the aforementioned types of setting, the primary objective of matrix completion involves predicting missing entries.

Under this type of setting, a natural approach for this involves minimizing the rank of the matrix under constraints yielded by observed entries, and this is generally referred to as "low-rank matrix completion". Unfortunately, it is NP-hard to literally solve the rank minimization problem. In order to practically overcome the difficulty, relaxation of matrix rank to nuclear norm was proposed [30]. Interestingly, it is guaranteed that the solution of the nuclear norm minimization is exactly in agreement with that of the original rank minimization if certain conditions are satisfied [31, 32, 33, 34]. The minimization of nuclear norm belongs to the class of convex optimization problems, and thus the optimal solution is determined via versatile semidefinite programming solvers when the matrix size is relatively small. However, in several realistic problems, matrix sizes are not so small, and computational and memory costs required by the nuclear norm minimization often exceed practically acceptable levels.

In order to deal with such situations, a non-convex approach using matrix factorization was proposed more recently [29]. When the objective matrix is factorized into two matrices of lower rank, nuclear norm is evaluated as the sum of their Frobenius norms. The non-convex formulation significantly reduces necessary computational and memory costs while we can generally find only local minima. However, a recent study [35] indicated that under a certain condition, the objective function of matrix factorization does not exhibit spurious local minima. Each local minimum is transformed to another via trivial operations such as permutations of column/rows with high probabilities.

In this section, we discuss the several types of the matrix completion problems: rank minimization, nuclear norm minimization, and matrix factorization. Together with these, standard algorithms to solve them will be introduced.

2.3.1 Rank minimization

As before, the goal of matrix completion is to recover missing entries of sparse observed matrix Y. To this end, we often assume that Y is a low-rank matrix. If the additional constraint is not imposed, it is an ill-posed problem, and we can fill the missing entries with arbitrary values. The low-rank constraint originates from a practical situation that the amount of principal information contained in the observed matrix is often much smaller than its data size, as mentioned in Introduction. To see this in more detail, let us consider the Netflix prize [36] as a famous example of the application of the low-rank matrix completion. The Netflix prize was the competition of algorithms for the recommender system, and particularly focused on user-movie rating datasets. Users who watched a movie rate it by five-point scales; if the movie is preferred by the user, it gets score 5, while if it is not, it gets score 1. Here, in observed matrix $Y \in \mathbb{R}^{N\times M}$, where N is the number of users and M is the number of movies, its (μ, i)th element $y_{\mu i} \in \{1, 2, 3, 4, 5\}$ denotes a rating from user μ to movie i. However, users normally evaluate only a small fraction of movies, and thus most entries of Y are missing. The primary objective of matrix completion involves predicting missing entries. Under the aforementioned types of setting, the low-rank approximation for Y is reasonable. This is because of the underlining assumption of the collaborative filtering, which states that users who have similar preferences tend to watch similar movies. Here, the number of the users'

preferences approximately correspond to the rank of Y. Therefore, the low-rank assumption is reasonable because the number of the users' preferences is normally much smaller than N. Similarly, the genres of movies also correspond to the matrix rank. In the case of the Netflix prize, although Y is a significantly large matrix ($N = 480,189$ and $M = 17,770$), the amount of its principal information is regarded to be much smaller because of the low-rank approximation.

First, we introduce the low-rank matrix completion problem via the rank minimization. It is defined as

$$\begin{aligned} \min_{X} \quad & \operatorname{rank}(X) \\ \text{subject to} \quad & x_{\mu i} = y_{\mu i}, \quad (\mu, i) \in \Omega \end{aligned} \tag{2.54}$$

where X is the decision matrix, $y_{\mu i}$ is an element of the observed matrix, and Ω is the set of observed entries. This problem is to find the lowest rank matrix under the constraint that the elements of the decision matrix are the same as the observed elements. However, this problem is known to be NP-hard.

The above problem was defined in a noise-free environment. However, it is far from a practical setting in general. Therefore, we next consider the low-rank matrix completion in a noisy environment [37]. This problem is defined as

$$\begin{aligned} \min_{X} \quad & \operatorname{rank}(X) \\ \text{subject to} \quad & \sqrt{\sum_{(\mu,i)\in\Omega} (x_{\mu i} - y_{\mu i})^2} \leq \delta, \end{aligned} \tag{2.55}$$

where $\delta > 0$ is a tolerance parameter for the fitting error. Overfitting against the noise can be prevent depending on the tolerance parameter. Further, we can recast Eq. (2.55) as

$$\begin{aligned} \min_{X} \quad & \sum_{(\mu,i)\in E} (x_{\mu i} - y_{\mu i})^2 \\ \text{subject to} \quad & \operatorname{rank}(X) \leq r, \end{aligned} \tag{2.56}$$

Here, r is also a tolerance parameter, which corresponds to δ in Eq. (2.55). When Y is a full-filled observed matrix, the optimal solution of the minimization problem (2.56) is given by the rank-r SVD of Y. However, when Y has missing entries, it is difficult to find the optimal solution because the problem becomes non-convex. In this case, if giving up on finding the global minimum, a handy algorithm based on heuristics can be used to find a local minimum. First, missing entries of Y are filled in with zero to prepare an initialization matrix. Second, the rank-r SVD is performed for the filled matrix, and then only the entries that were missing are replaced by the corresponding entries of the rank-r approximation matrix. Here, the observed entries remain unchanged. These procedures are repeated until convergence is achieved. This heuristic method requires the rank-r SVD computation at each iteration. Thus, the computational cost at each iteration is $O(N^2)$. This computational cost is too high to deal with real data in many practical cases. Therefore, to transform into a more tractable problem, we introduce some relaxations and assumptions in the following subsection.

2.3.2 Nuclear norm minimization

The nuclear norm minimization is a common approach to transform the rank minimization problem into a more tractable convex problem. Because the rank function is not convex and difficult to handle flexibly, we consider to approximate it as a more tractable form. To this end, the nuclear norm $\|X\|_*$ has suitable properties. It is defined as

$$\|X\|_* = \sum_{k=1}^{\min\{N,M\}} \sigma_k, \tag{2.57}$$

where $\sigma_k \geq 0$ is the kth largest singular value of X. The rank of X is determined by the number of non-zero singular values of X. On the other hand, the nuclear norm is defined as the sum of singular values of X. Therefore, it can be expected that the nuclear norm minimization is a good approximation of the rank minimization. By replacing the matrix rank in Eq. (2.54) with the nuclear norm, we obtain

$$\begin{aligned} \min_X \quad & \|X\|_* \\ \text{subject to} \quad & x_{\mu i} = y_{\mu i}. \quad (\mu, i) \in \Omega \end{aligned} \tag{2.58}$$

In addition, the rank minimization problem in noisy environment (Eqs. (2.55) and (2.56)) can also be relaxed to

$$\min_X \frac{1}{2} \sum_{(\mu,i)\in\Omega} (y_{\mu i} - x_{\mu i})^2 + \lambda \|X\|_* \ . \tag{2.59}$$

Here, we describe it in Lagrangian form. $\lambda > 0$ is a constant parameter for inducing the reduction of $\|X\|_*$. It is chosen appropriately by using such as cross-validation.

Now we wonder if the solution from the nuclear norm minimization problem matches that of the original rank minimization problem. A number of studies have tackled this problem and shown that solving either problem can recover the original matrix if several conditions are satisfied. These conditions are related to the so-called coherence and the number of observed entries. They are briefly described below.

Coherence is intuitively a measure of how biased the matrix entries are. The higher the coherence is, the more biasedly distributed the matrix entries are, and the more difficult it is to recover the missing entries. As an example, let us consider a rank-1 matrix as follows.

$$S_1 = \boldsymbol{e}_N \boldsymbol{e}_1^T = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & \cdots & 0 & 0 \end{pmatrix}, \tag{2.60}$$

where $\boldsymbol{e}_i$ is the basis vector which have all zero entries except that its ith element is 1. Although recovering missing entries of a rank-1 matrix is a relatively easy problem in general, this case is not easy. This is because this matrix has only one non-zero entry in the bottom-left corner, and we cannot hope to recover the non-zero entry unless it is observed directly. This is an example of a matrix that has much high coherence. On the other hand, as an

example of a low coherence rank-1 matrix, we can consider $S_2 = \boldsymbol{u}_1\boldsymbol{u}_2^T$, where $\boldsymbol{u}_1$ and $\boldsymbol{u}_2$ are random vectors whose entries are independently sampled from the standard Gaussian distribution. We can expect that the entries of the random matrix are not biased rather than S_1, and its recovery can be easier. This indicates that the coherence is closely related to the difficulty to recover the missing entries.

In addition, the number of the observed entries is also an essential factor for the condition to recover the missing entries. Of course, a mostly filled observed matrix is easier to be recovered, and it becomes more difficult as the number of observations decreases. By focusing on the degree of freedom of the rank-R matrix, we can confirm that at least $(N+M)R-2R^2$ observations are required. This can be verified by considering the rank-R SVD. The second term comes from the constraints of the orthogonal matrices.

From another point of view, it turns out that at least $O(N \log N)$ observations are required for the recovery. This comes from the so-called coupon collector's problem [38]. The definition of this problem is simple. Let us assume a situation that there are a large number of coupons of which have N types, and we take out the coupons one by one randomly and independently to check the content of the coupon. In this situation, how many trials on average to collect all types of coupons is given by $O(N \log N)$. This is closely related to the number of observations to recover the missing entries. Let us consider a rank-1 matrix Y_1 as an example as follows.

$$Y_1 = \begin{pmatrix} * & * & \cdots & * & * \\ y_{21} & y_{22} & \cdots & y_{2(N-1)} & y_{2N} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{N(N-1)} & y_{NN} \end{pmatrix}. \tag{2.61}$$

Here, $*$ represents a missing entry and all entries of the first row of Y_1 are missing. In this case, we cannot hope to recover the first row. This is because when we can factorize as $Y_1 = \boldsymbol{v}_1\boldsymbol{v}_2^T$, where $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ are a certain vector, we have no information to infer the first element of $\boldsymbol{v}_1$. Accordingly, we require at least one observation in a row and column. This can be viewed as the coupon collector's problem. In the following, we review the coupon collector's problem shortly and how the fact the $O(N \log N)$ observations are required is derived.

First, we define Z as a random variable following the geometric distribution with parameter p, i.e., $P(Z = k) = (1-p)^{k-1}p$. Thus, the random variable of the geometric distribution indicates the number of trials required before an event with probability p occurs. Using this, the expected number of trials to collect all types of coupons can be expressed as

$$\mathbb{E}[X] = \mathbb{E}[X_1] + \cdots + \mathbb{E}[X_N], \tag{2.62}$$

where X_i is a random variable following the geometric distribution with parameter $(N-i)/N$. This is because when i types of coupons have been already collected, the probability to hit an uncollected coupon by one trial is $(N-i)/N$. Therefore, using $X_1 = 1$ and $\mathbb{E}[X_i] = N/(N-i)$, we obtain

$$\mathbb{E}[X] = 1 + \frac{N}{N-1} + \frac{N}{N-2} + \cdots + \frac{N}{2} + N \tag{2.63}$$

$$= NH_N, \tag{2.64}$$

where $H_N = \sum_{i=1}^{N}(1/i)$ is the harmonic number. According to the asymptotic expansion of the harmonic number, we obtain

$$H_N = \log N + \gamma + O\left(\frac{1}{N}\right), \tag{2.65}$$

where $\gamma \approx 0.57721$ is the Euler–Mascheroni constant. Accordingly, when N is large, we can confirm the expected number of trials is given by $\mathbb{E}[X] = O(N \log N)$.

A number of studies have theoretically investigated the number of the observations needed to recover the original matrix perfectly by solving the nuclear norm minimization problem. When the original matrix has sufficiently small coherence, Candès and Recht (2008) [31] shows $M \geq CN^{1.2}R \log N$ observations are required to recover the original matrix with high probability. C is a positive constant. After that, several studies shows the more tight inequality [32, 33, 34, 39].

To solve the nuclear norm minimization problem, we have many options because it is convex unlike the rank minimization problem. In a noise-free environment (2.58), Fazel (2002) [30] shows the nuclear norm minimization problem can be recast as an equivalent problem in terms of semidefinite programming. This is derived by using the fact that the nuclear norm is expressed by the sum of the singular values. A number of efficient methods can be used to solve such types of problem [40]. However, standard solvers for such problems are problematic when the system size is large. To alleviate it, a more scalable method has been proposed in Cai at el (2008) [41]. The proposed method, which is referred to as singular value thresholding algorithm, is a first-order method, and it is scalable for the system size and adaptable to a larger matrix. This method consists of two steps. The first step shrinks the singular values to induce a low rank, and the second step updates the variable to fit the observed entries. In the first step, the shrinkage is performed by using the singular value shrinkage operator $\mathcal{S}_\lambda$, which is defined as

$$\mathcal{S}_\lambda(X) = U\Sigma_\lambda V^T, \tag{2.66}$$

where $\Sigma_\lambda = \mathrm{diag}[(\sigma_1 - \lambda)_+, \cdots, (\sigma_R - \lambda)_+]$. Here, $(x)_+$ represents $x = 0$ if $x < 0$, and U and V are given by the rank-R SVD $X = U\Sigma V^T$. The operator $\mathcal{S}_\lambda(X)$ works to shrink the rank of matrix X by λ. The singular value shrinkage operator is closely related to the nuclear norm minimization problem. It obeys

$$\mathcal{S}_\lambda(Z) = \arg\min_X \left\{ \frac{1}{2}\|X - Z\|_F^2 + \lambda \|X\|_* \right\}. \tag{2.67}$$

The proof to show this equation can be found in Ref. [41]. Note that the right-hand side of Eq. (2.67) is defined for a dense matrix Z. Using this operator, the following heuristic procedure can be considered

$$\begin{cases} Z^k = \mathcal{S}_\lambda\left(X^{k-1}\right) \\ X^k = X^{k-1} + \delta_k \mathcal{P}_\Omega\left(Y - Z^k\right), \end{cases} \tag{2.68}$$

where $\mathcal{P}_\Omega$ is the projection operators, which is defined as

$$\left[\mathcal{P}_\Omega(Y)\right]_{ij} = \begin{cases} y_{ij} & \text{if } (i,j) \in \Omega \\ 0 & \text{if } (i,j) \notin \Omega. \end{cases} \tag{2.69}$$

Thus, the role of operator $\mathcal{P}_\Omega(Y)$ is to fill the missing entries of Y to zero. δ_k is the step size in kth iteration, and Y is the observation matrix. This procedure starts with $X^0 = \mathbf{0}$ and is repeated for $k = 1, 2, \ldots$ until convergence. This convergence is guaranteed and its proof is presented in Ref. [41]. Of course, because the rank of X is unknown in advance, we choose relatively large R ($< M$). However, its redundant rank can reduce to 0 by the operator $\mathcal{S}_\lambda$ according to λ.

Mazumber at el (2010) [42] propose another algorithm, which is called SOFT-IMPUTE, for the nuclear norm minimization. This algorithm is inspired by the heuristic algorithm for a rank minimization problem, which was introduced in Sec. 2.3.1. This heuristic algorithm performs the rank-R SVD repeatedly and updates only the entries corresponding to the missing ones. Similarly to this, SOFT-IMPUTE operates the singular value shrinkage operator repeatedly and updates only the entries corresponding to the missing ones. Its specific procedure is given as follows.

$$X^k = \mathcal{S}_{\lambda_k}\left(\mathcal{P}_\Omega(Y) + \mathcal{P}_\Omega^\perp(X^{k-1})\right). \tag{2.70}$$

Here, $\mathcal{P}_\Omega^\perp(X)$ is an operator to fill the (i,j)th entry of X to zero, where $(i,j) \notin \Omega$, and λ_k is predetermined in advance by a decreasing grid, i.e., $\lambda_1 > \lambda_2 > \cdots$. The procedure starts with $X^0 = \mathbf{0}$ and is repeated for $k = 1, 2, \ldots$ until convergence. This algorithm also guaranteed to converge to the global minimum. Besides, it has an advantage rather than the previous algorithm in term of the computational cost. The input of the operator $\mathcal{S}_{\lambda_k}$ in Eq. (2.70) can be reformulate as

$$S^{k-1} = \mathcal{P}_\Omega(Y) + \mathcal{P}_\Omega^\perp(X^{k-1}) = \mathcal{P}_\Omega(Y - X^{k-1}) + X^{k-1}. \tag{2.71}$$

Here, the first term in the rightmost side is a sparse matrix, and the second term is a low-rank matrix. The leading computational cost in Eq. (2.70) comes from the calculation of the rank-R SVD, whose computational cost is generally $O(MNR)$ by using standard algorithms such as a Lanczos method [8]. However, we can use a specific structure in Eq. (2.71) to reduce the computational cost to be linear with the matrix size. The Lancroz method is based on the powered method, and its most expensive step stems from matrix-vector multiplications, i.e., $\boldsymbol{u}^T S^{k-1}$ or $S^{k-1}\boldsymbol{v}$, where $\boldsymbol{u} \in \mathbb{R}^N$ and $\boldsymbol{v} \in \mathbb{R}^M$. By multiplying $\boldsymbol{v}$ from the right in Eq. (2.71), we obtain

$$S^{k-1}\boldsymbol{v} = \mathcal{P}_\Omega(Y - X^{k-1})\boldsymbol{v} + X^{k-1}\boldsymbol{v}. \tag{2.72}$$

The computational cost of the first term in the right-hand side is $O(|\Omega|)$ because we can ignore the computation of the zero entries, and the computational cost of the second term is $O((N + M + R)R)$ because it can be factorized by the rank-R SVD as $X^{k-1} = U^{k-1}\Sigma^{k-1}(V^{k-1})^T$. Here, this rank-$R$ SVD has already been computed in the previous iteration. This is analogous to the left multiplication $\boldsymbol{u}^T S^{k-1}$. Therefore, when a small R ($\ll N, M$) is chosen, the computational cost is linear with the matrix size. In addition, the memory cost is also significantly reduced due to the sparse and low-rank structures.

The SOFT-IMPUTE succeeds in reducing the computational cost significantly. However, it requires to compute the rank-R SVD for each iteration. In a practical setting, the observed matrix size is extremely large in many cases. Therefore, the alleviation of the computational cost is not enough. To address the practical problems, we introduce a further relaxation below.

2.3.3 Low-rank matrix factorization

The methods to solve the nuclear norm minimization problem required to compute the rank-R SVD for each iteration. To alleviate the high computational cost for practical use, we introduce a low-rank matrix factorization technique in this subsection. As seen before, the low-rank matrix factorization is a common technique for dimensionality reduction in a wide range of fields. Given a rank-R matrix $X \in \mathbb{R}^{N\times M}$, it can be factorized into two smaller matrices $U \in \mathbb{R}^{N\times R}$ and $V \in \mathbb{R}^{M\times R}$, i.e., $X = UV^T$. The rank-R SVD is a specific case of the low-rank matrix factorization, where the orthogonality is imposed among rows in the factorized matrices. Due to the constraints, the rank-R SVD provides an unique factorization when the order of its singular values is fixed. On the other hand, the factorization cannot be determined uniquely without any constraints. For arbitrary orthogonal matrix $O \in \mathbb{R}^{R\times R}$, we can recast as $X = UV^T = UO(VO)^T$, namely they have the rotational symmetry. However, the lack of the orthogonal constraints and uniqueness is often less of an issue, rather it has practical advantages in matrix completion problems. First, when any constraints are imposed, it can be solved faster because some efficient iterative algorithms such as stochastic gradient decent (SGD) and alternating least square (ALS) can be used. Second, in terms of the matrix completion problems, the uniqueness of the factorized matrices are not required as long as the missing entries can be recovered uniquely. Here, we define the matrix completion problem using the low-rank matrix factorization as follows.

$$\min_{U,V} \frac{1}{2} \sum_{(\mu,i)\in\Omega} \left(y_{\mu i} - \sum_{r=1}^{R} u_{\mu r} v_{ir} \right)^2 + \frac{1}{2}\lambda \|U\|_F^2 + \frac{1}{2}\lambda \|V\|_F^2 \ . \tag{2.73}$$

Here, the second and third terms are the regularization terms, and λ controls the extent of shrinkage for the variables. As discussed later, they originate from the nuclear norm. Actually, this problem cannot guarantee the uniqueness of the recovered entries because it is no longer a convex problem. However, it was empirically known that the recovered matrix obtained from solving Eq. (2.73) matches the original matrix very well. Given this fact, Ge at el. [35] provides a proof that every local minimum to which the gradient decent algorithm converges is also a global minimum even if it starts from a random initialization. Therefore, the formulation of the low-rank matrix factorization is suitable for practical matrix completion problems. However, it should be noted that the rank of the factorized matrices must be determined in advance, and it is generally kept unchanged, unlike the nuclear norm minimization. In the following, we will see the relation to the nuclear norm minimization problem, and introduce some existing major algorithms to optimize Eq. (2.73).

The low-rank matrix factorization problem (2.73) is closely related to the nuclear norm minimization problem (2.59) when the matrix rank is given by R. According to Refs. [43, 42, 44], the nuclear norm is expressed by the sum of the Frobenius norms of the factorized matrices as follows:

$$\|X\|_* = \min_{U,V} \left\{ \frac{1}{2} \left(\|U\|_F^2 + \|V\|_F^2 \right) \ : \ X = UV^T \right\}. \tag{2.74}$$

This equation can be verified as follows. Letting the rank-R SVD of X be $X = U_R \Sigma_R V_R^T$,

the nuclear norm $\|X\|_*$ can be rewritten as follows.

$$\|X\|_* = \mathrm{Tr}(\Sigma_R) \tag{2.75}$$
$$= \mathrm{Tr}(U_R^T X V_R) \tag{2.76}$$
$$= \mathrm{Tr}\left(U_R^T U \left(V_R^T V\right)^T\right). \tag{2.77}$$

Here, we can derive the following relationship between the trace and the Frobenius norms for two real matrices B and C.

$$\mathrm{Tr}(BC^T) = \sum_{ij} b_{ij} c_{ij} \tag{2.78}$$
$$\leq \left(\sum_{ij} b_{ij}^2\right)^{1/2} \left(\sum_{ij} c_{ij}^2\right)^{1/2} \tag{2.79}$$
$$= \|B\|_F \|C\|_F. \tag{2.80}$$

To derive Eq. (2.79), we used the Cauchy–Schwarz inequality. Using this inequality for Eq. (2.77), we obtain

$$\|X\|_* \leq \|U_R^T U\|_F^2 \|V_R^T V\|_F^2 \tag{2.81}$$
$$= \|U\|_F^2 \|V\|_F^2 \tag{2.82}$$
$$\leq \frac{1}{2}\left(\|U\|_F^2 + \|V\|_F^2\right). \tag{2.83}$$

Here, we used relation $(\|U\|_F - \|V\|_F)^2 \geq 0$ and the fact that the Frobenius norm is invariant for orthogonal transformations. We can confirm that the equalities are accomplished when $U = U_R \Sigma_R^{1/2}$ and $V = V_R \Sigma_R^{1/2}$. By inserting Eq. (2.74) into Eq. (2.59), we obtain Eq. (2.73).

Problems like Eq. (2.73) are called the biconvex problem because it reduces to a convex problem if one of the two types of variable U and V is fixed. To solve these types of problem, a number of efficient local search algorithms can be used. These algorithms can work fast and have scalability for significantly large scale datasets. In the following, we introduce two major algorithms to solve Eq. (2.73): alternating least squares (ALS) and stochastic gradient decent (SGD).

The ALS is a widely known approach to the biconvex optimization problem due to its simplicity. A number of variants have been proposed and their theoretical analyses are provided [45, 46, 47, 48]. When V is fixed, each row of U is independently calculated, and the minimization problem of (2.73) is then expressed as follows.

$$\min_{\mathbf{u}_\mu} \frac{1}{2} \sum_{i \in \partial\mu} \left(y_{\mu i} - \mathbf{u}_\mu^T \mathbf{v}_i\right)^2 + \lambda \left\|\mathbf{u}_\mu\right\|^2, \tag{2.84}$$

where $\boldsymbol{u}_\mu$ and $\boldsymbol{v}_i$ denote the μth and ith rows of U and V respectively, and $\partial\mu$ denotes a set of observed indices of μth row of Y. Thus, Eq. (2.84) leads to the following closed form solution

$$\mathbf{u}_\mu^* = \left(\sum_{i \in \partial\mu} \mathbf{v}_i \mathbf{v}_i^T + \lambda \mathbf{I}_R\right)^{-1} \left(\sum_{i \in \partial\mu} y_{\mu i} \mathbf{v}_i\right), \tag{2.85}$$

where $\boldsymbol{I}_R$ denotes $R \times R$ unit matrix. Subsequently, we fix U and solve V in tern, and ALS repeats this operation until convergence. The main advantage of ALS is the ease of the parallelization and distribution. This is realized because its update rules are described independently with respect to each row of U and V, like Eq. (2.85). Another advantage of ALS is that because its update equation (2.85) can be described as a closed form, it is expected to converge faster and does not require the delicate scheduling of any additional parameters such as a learning rate. On the other hand, as its disadvantage, its computational costs are more expensive because of the inverse matrix in Eq (2.85). Note that hereafter, we use 'iteration' for counting the unit of computation in which all variables are updated once on average, while the unit of elemental renewal of the variables is referred to as 'update'.

The other algorithm, SGD, is also widely known as a standard algorithm for continuous optimization problem. Specifically, SGD computes a gradient only with respect to pairwise indices $(\mu, i) \in \Omega$ selected at random, and the gradient renews the corresponding variables based on the given learning rate η as follows.

$$\mathbf{u}_\mu \leftarrow \mathbf{u}_\mu - \eta \left\{ \lambda \mathbf{u}_\mu - \left(y_{\mu i} - \mathbf{u}_\mu^T \mathbf{v}_i \right) \mathbf{v}_i \right\}, \tag{2.86}$$

$$\mathbf{v}_i \leftarrow \mathbf{v}_i - \eta \left\{ \lambda \mathbf{v}_i - \left(y_{\mu i} - \mathbf{u}_\mu^T \mathbf{v}_i \right) \mathbf{u}_\mu \right\}. \tag{2.87}$$

This algorithm exhibits an advantage wherein its computational cost per update is lower. However, it has two major disadvantages. The first is that an overwriting issue can arise when the several updates are conducted in parallel. The second is that it is highly sensitive to the learning rate. Distributed SGD (DSGD) [49] (the name *Jellyfish* used in [49]) overcomes the first disadvantage by dividing the observed matrix into a few blocks, considering a set of independent blocks, and updating a pair of indices from each block in it. However, the second disadvantage still remains, and the learning rate should be carefully tuned and scheduled. The adjustment of the learning rate significantly affects the convergence of the algorithm.

In Chapter 3, we develop algorithms, which we call cavity-based matrix factorization (CBMF) and approximate cavity-based matrix factorization (ACBMF), borrowing an idea from the cavity method of statistical mechanics. They are performable with low computational costs in parallel and/or distributed manners.

Chapter 3

Approximate matrix completion based on cavity method

As mentioned in the end of the previous chapter, two major algorithms, alternating least squares (ALS) [50, 45, 46] and stochastic gradient descent (SGD) [51, 52, 49] are proposed for the matrix factorization to date. The main objective of this study is to develop a new algorithm by borrowing an idea from the cavity method from statistical mechanics. Even if the absence of spurious local minima is guaranteed, the performance of the solution search is determined via dynamical properties of the used algorithm. We experimentally illustrate that the proposed cavity-based algorithms exhibit better performance than the two algorithms without delicate tuning of control parameters when the number of observed data is relatively small.

Several existing studies apply the cavity method for the matrix factorization problems. An approximate message passing (AMP) based approach to generalized bilinear inference problem including the matrix completion was proposed in Ref. [53, 54]. A detailed derivation of AMP-type algorithms and performance analysis for the Bayes optimal cases are provided in Ref. [55]. Reference [56] presents an AMP based algorithm for low-rank matrix reconstruction and its application to K-means type clustering. All of these methods follow the Bayesian framework. The differences of the present study from these are as follows. We do not employ the Bayesian approach, and thus it is not necessary to select a prior distribution. Additionally, we focus on the matrix completion as a particular application of matrix factorization, and aim to develop efficient algorithms exploiting the properties of the specific problem.

The remainder of this chapter is organized as follows. In Sec. 3.1, we explain the details of the proposed algorithm. In Sec. 3.2, the performance of the proposed algorithms is illustrated via applications for synthetic and realistic data. The final section presents the summary.

3.1 A Cavity-Based Approach

In order to explore the possibility of achieving a better performance, we develop an algorithm for the matrix factorization based on the cavity method [58]. Thus, we first express the variable dependence of Eq. (2.73) by a factor graph (Fig. 3.1). The variable

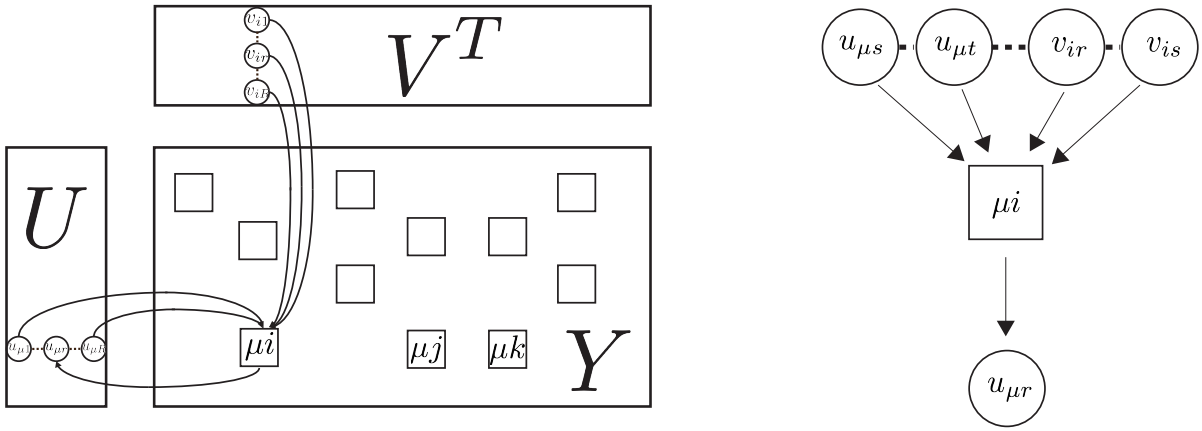


Figure 3.1: Graphical expression of (3.1) (left) and its enlarged illustration (right). Circle and squares correspond to variable and function nodes, respectively. Equation (3.2) is also computed in a similar manner. The figures are taken from Ref. [57].

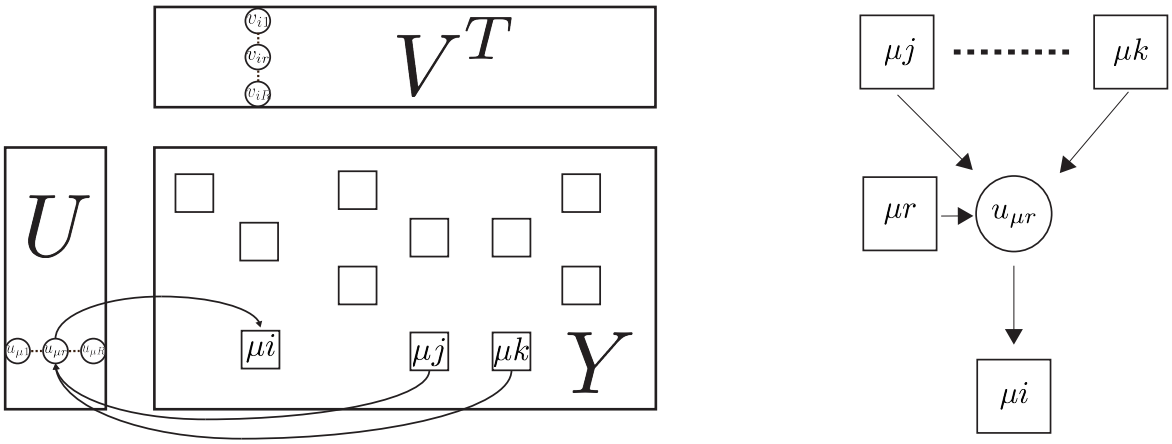


Figure 3.2: Graphical expression of (3.3) (left) and its enlarged illustration (right). Equation (3.4) is also computed in a similar manner. The figures are taken from Ref. [57].

nodes are expressed by circles and denote entries of two matrices U and V while the factor nodes are represented by squares and stand for factors constituting (2.73), namely, $(1/2)\left(y_{\mu i}-\sum_{r=1}^{R}u_{\mu r}v_{ir}\right)^2$,$(\lambda/2)u_{\mu r}^2$ and $(\lambda/2)v_{ir}^2$. An edge for a pair of variable and factor nodes is provided if and only if the variable and factor nodes are directly related.

The basic idea of the cavity method is to approximate the multivariate minimization problem (2.73) via a bunch of minimization problems with respect to single variables. Hence, we introduce "cavity objective functions" $f_{\mu r\to(\mu i)}(u_{\mu r})$ and $g_{ir\to(\mu i)}(v_{ir})$. The function $f_{\mu r\to(\mu i)}(u_{\mu r})$ denotes the objective function after the minimization with respect to all variables other than $u_{\mu r}$ is performed in the "(μi)-cavity system" that is defined by removing $(1/2)\left(y_{\mu i}-\sum_{r=1}^{R}u_{\mu r}v_{ir}\right)^2$ from Eq. (2.73), and similarly for $g_{ir\to(\mu i)}(v_{ir})$. The summation of the cavity objective functions and $(1/2)\left(y_{\mu i}-\sum_{r=1}^{R}u_{\mu r}v_{ir}\right)^2$ approximates the full objective function of Eq. (2.73). Conversely, we remove the contribution of $f_{\mu r\to(\mu i)}(u_{\mu r})$ from the full summation and minimize the resulting function with respect to all variables except for $u_{\mu r}$. This yields "cavity bias function" $\hat{f}_{(\mu i)\to\mu r}(u_{\mu r})$, and this denotes the effective influence of the factor $(1/2)\left(y_{\mu i}-\sum_{r=1}^{R}u_{\mu r}v_{ir}\right)^2$ to the variable $u_{\mu r}$, and similarly for $\hat{g}_{(\mu i)\to ir}(v_{ir})$. The summation of the cavity bias functions with the exception of $\hat{f}_{(\mu i)\to\mu r}(u_{\mu r})$ and $(\lambda/2)u_{\mu r}^2$ yields $f_{\mu r\to(\mu i)}(u_{\mu r})$, and similarly for $g_{ir\to(\mu i)}(v_{ir})$. They constitute a closed set of functional equations to determine the cavity objective and bias functions as follows:

$$\hat{f}_{(\mu i)\to\mu r}(u_{\mu r})=\min_{\{\mathbf{u}_\mu,\mathbf{v}_i\}\backslash u_{\mu r}}\left\{\frac{1}{2}(y_{\mu i}-\sum_s u_{\mu s}v_{is})^2+\sum_{s\neq r}f_{\mu s\to(\mu i)}(u_{\mu s})+\sum_s g_{is\to(\mu i)}(v_{is})\right\}, \tag{3.1}$$

$$\hat{g}_{(\mu i)\to ir}(v_{ir})=\min_{\{\mathbf{u}_\mu,\mathbf{v}_i\}\backslash v_{ir}}\left\{\frac{1}{2}(y_{\mu i}-\sum_s u_{\mu s}v_{is})^2+\sum_s f_{\mu s\to(\mu i)}(u_{\mu s})+\sum_{s\neq r}g_{is\to(\mu i)}(v_{is})\right\}, \tag{3.2}$$

$$f_{\mu r\to(\mu i)}(u_{\mu r})=\sum_{(\mu j)\in\partial\mu r\backslash(\mu i)}\hat{f}_{(\mu j)\to\mu r}(u_{\mu r})+\frac{1}{2}\lambda u_{\mu r}^2, \tag{3.3}$$

$$g_{ir\to(\mu i)}(v_{ir})=\sum_{(\nu i)\in\partial ir\backslash(\mu i)}\hat{g}_{(\nu i)\to ir}(v_{ir})+\frac{1}{2}\lambda v_{ir}^2, \tag{3.4}$$

where $\mathbf{u}_\mu$ and $\mathbf{v}_i$ denote the μ-th and i-th rows of U and V, respectively, and $A\backslash a$ generally indicates a set that is defined via eliminating an element a from a set A. The indices of factor nodes are denoted with parentheses while those of variable nodes are not. The notation $\partial\mu r$ stands for the set of factor nodes that directly connect variable node indexed by μr. After determining the cavity objective and bias functions from Eqs. (3.1)-(3.4), "marginal" objective functions for each variable are provided as follows:

$$f_\mu(u_{\mu r})=\sum_{(\mu i)\in\partial\mu r}\hat{f}_{(\mu i)\to\mu r}(u_{\mu r})+\frac{1}{2}\lambda u_{\mu r}^2, \tag{3.5}$$

$$g_i(v_{ir}) = \sum_{(\mu i)\in\partial ir} \hat{g}_{(\mu i)\to ir}(v_{ir}) + \frac{1}{2}\lambda v_{ir}^2. \tag{3.6}$$

Thus, entries of the factorized matrices are evaluated as follows:

$$u_{\mu r}^* = \arg\min_{u_{\mu r}} \left\{ f_\mu(u_{\mu r}) \right\}, \tag{3.7}$$

$$v_{ir}^* = \arg\min_{v_{ir}} \left\{ g_i(v_{ir}) \right\}. \tag{3.8}$$

3.1.1 Derivation of the algorithm

Two issues are emphasized here. First, when the factor graph does not contain any cycles, the solution given by the cavity method is exact. However, cycles generally exist in the matrix factorization problem. However, if the positions of the observed entries are randomly selected and their number is limited up to $O(N)$ as assumed in the following, then the resulting factor graph is considered as a sparse random graph. Thus, the lengths of the cycles typically scale as $O(\ln N)$ when the system size N increases. Therefore, it is reasonable to expect that the cavity method yields reasonably accurate approximates for large N as the effect of the cycles becomes negligible. Second, solving Eqs. (3.1)-(3.4) is, unfortunately, technically difficult since they are provided as functional equations. In order to overcome the difficulty, we parameterize the cavity objective and bias functions in the form of quadratic functions as follows:

$$\begin{aligned}
\hat{f}_{(\mu i)\to\mu r}(u_{\mu r}) &= \frac{1}{2}\hat{a}_{(\mu i)\to\mu r} u_{\mu r}^2 - \hat{b}_{(\mu i)\to\mu r} u_{\mu r}, & (3.9)\\
\hat{g}_{(\mu i)\to ir}(v_{ir}) &= \frac{1}{2}\hat{c}_{(\mu i)\to ir} v_{ir}^2 - \hat{d}_{(\mu i)\to ir} v_{ir}, & (3.10)\\
f_{\mu r\to(\mu i)}(u_{\mu r}) &= \frac{1}{2} a_{\mu r\to(\mu i)} u_{\mu r}^2 - b_{\mu r\to(\mu i)} u_{\mu r} + \frac{1}{2}\lambda u_{\mu r}^2, & (3.11)\\
g_{ir\to(\mu i)}(v_{ir}) &= \frac{1}{2} c_{ir\to(\mu i)} v_{ir}^2 - d_{ir\to(\mu i)} v_{ir} + \frac{1}{2}\lambda v_{ir}^2. & (3.12)
\end{aligned}$$

However, the insertion of Eqs. (3.9)-(3.12) into Eqs. (3.1)-(3.4) does not yield a closed form of equations to determine the parameters. This indicates that a further approximation is required. For this, in optimizing Eq. (3.1), we fix $\mathbf{v_i}$ to the value in the previous iteration. This yields quadratic form with respect to $\mathbf{u}_\mu^r$ and leads to a set of closed form equations, where $\mathbf{u}_\mu^r$ denotes a vector excluding $u_{\mu r}$ from $\mathbf{u}_\mu$. More explicitly, Eq. (3.1) is then reduced to

$$\hat{f}_{(\mu i)\to\mu r}(u_{\mu r}) = \min_{\{\mathbf{u}_\mu\}\setminus u_{\mu r}} \left\{ \frac{1}{2}(\mathbf{u}_\mu^r)^T \left(\Gamma_{\mathbf{a}_{\mu\to(\mu i)}^r} + \mathbf{v}_i^r (\mathbf{v}_i^r)^T \right) \mathbf{u}_\mu^r - \left\{ \mathbf{b}_{\mu\to(\mu i)}^r + (y_{\mu i} - u_{\mu r} v_{ir}) \mathbf{v}_i \right\}^T \mathbf{u}_\mu^r \right\}, \tag{3.13}$$

where $\mathbf{a}^r_{\mu\to(\mu i)}$ denotes a vector excluding $a_{\mu r}$ from $\mathbf{a}_{\mu\to(\mu i)} = (a_{\mu 1\to(\mu i)}, ..., a_{\mu R\to(\mu i)})$, and $\Gamma_{\mathbf{a}^r_{\mu\to(\mu i)}}$ and $\Gamma_{\mathbf{b}^r_{\mu\to(\mu i)}}$ indicate, respectively, $\mathrm{diag}(\mathbf{a}^r_{\mu\to(\mu i)} + \lambda\mathbf{1})$ and $\mathrm{diag}(\mathbf{b}^r_{\mu\to(\mu i)})$. Similarly for $\mathbf{b}^r_{\mu\to(\mu i)}$. A similar treatment is also made for Eq. (3.2) and $\mathbf{v}^r_i$, where $\mathbf{v}^r_i$ is a vector excluding v_{ir} from $\mathbf{v}_i$.

The minimization problem in Eq. (3.13) is solved as follows:

$$(\mathbf{u}^r_\mu)^* = \left(\Gamma_{\mathbf{a}^r_{\mu\to(\mu i)}} + \mathbf{v}^r_i(\mathbf{v}^r_i)^T\right)^{-1}\left\{\mathbf{b}^r_{\mu\to(\mu i)} - (y_{\mu i} - u_{\mu r}v_{ir})\mathbf{v}^r_i\right\}. \tag{3.14}$$

Based on Sherman–Morrison formula, the inverse matrix in (3.14) is re-expressed as follows:

$$(\Gamma_{\mathbf{a}^r_{\mu\to(\mu i)}} + \mathbf{v}^r_i(\mathbf{v}^r_i)^T)^{-1} = \Gamma^{-1}_{\mathbf{a}^r_{\mu\to(\mu i)}} - \frac{\Gamma^{-1}_{\mathbf{a}^r_{\mu\to(\mu i)}}\mathbf{v}^r_i(\mathbf{v}^r_i)^T\Gamma^{-1}_{\mathbf{a}^r_{\mu\to(\mu i)}}}{1 + (\mathbf{v}^r_i)^T\Gamma^{-1}_{\mathbf{a}^r_{\mu\to(\mu i)}}\mathbf{v}^r_i}. \tag{3.15}$$

We insert Eqs. (3.14) and (3.15) into Eq. (3.13) to yield the following expression:

$$\hat{f}_{(\mu i)\to\mu r}(u_{\mu r}) = \frac{1}{2}\frac{v^2_{ir}}{1 + \chi_{(\mu i)} - \frac{v^2_{ir}}{a_{\mu r\to(\mu i)}+\lambda}}u^2_{\mu r} - \frac{y_{(\mu i)} - \Delta_{(\mu i)} + u_{\mu r\to(\mu i)}v_{ir}}{1 + \chi_{(\mu i)} - \frac{v^2_{ir}}{a_{\mu r\to(\mu i)}+\lambda}}v_{ir}u_{\mu r}, \tag{3.16}$$

where $\chi_{(\mu i)}$, $\Delta_{(\mu i)}$ and $u_{\mu r\to(\mu i)}$ are defined as follows:

$$\chi_{(\mu i)} = \sum_r \frac{v^2_{ir}}{a_{\mu r\to(\mu i)} + \lambda}, \tag{3.17}$$

$$\Delta_{(\mu i)} = \sum_r u_{\mu r\to(\mu i)}v_{ir}, \tag{3.18}$$

$$u_{\mu r\to(\mu i)} = \frac{b_{\mu r\to(\mu i)}}{a_{\mu r\to(\mu i)} + \lambda}. \tag{3.19}$$

From Eqs. (3.9) and (3.16), we obtain the following:

$$\hat{a}_{(\mu i)\to\mu r} = \frac{v^2_{ir}}{1 + \chi_{(\mu i)} - \frac{v^2_{ir}}{a_{\mu r\to(\mu i)}+\lambda}}, \tag{3.20}$$

$$\hat{b}_{(\mu i)\to\mu r} = \frac{y_{(\mu i)} - \Delta_{(\mu i)} + u_{\mu r\to(\mu i)}v_{ir}}{1 + a_{(\mu i)} - \frac{v^2_{ir}}{\chi_{\mu r\to(\mu i)}+\lambda}}v_{ir}. \tag{3.21}$$

Further, we insert Eqs. (3.9) and (3.11) into Eq. (3.3) to yield the following expression:

$$a_{\mu r\to(\mu i)} = a_{\mu r} - \hat{a}_{(\mu i)\to\mu r}, \tag{3.22}$$

$$b_{\mu r\to(\mu i)} = b_{\mu r} - \hat{b}_{(\mu i)\to\mu r}, \tag{3.23}$$

where $a_{\mu r}$ and $b_{\mu r}$ are defined as follows:

$$a_{\mu r} = \sum_{(\mu i)\in\partial\mu r} \hat{a}_{(\mu i)\to\mu r}, \tag{3.24}$$

$$b_{\mu r} = \sum_{(\mu i)\in\partial\mu r} \hat{b}_{(\mu i)\to\mu r}. \tag{3.25}$$

Finally, entries of the factorized matrices $u^*_{\mu r}$ are re-expressed from Eq. (3.7) as follows:

$$u^*_{\mu r} = \frac{b_{\mu r}}{a_{\mu r}+\lambda} \tag{3.26}$$

Similarly, we can re-express equations with respect to $\hat{c}_{(\mu i)\to ir}, \hat{d}_{(\mu i)\to ir}$ and $c_{ir\to(\mu i)}, d_{ir\to(\mu i)}$ based on Eqs. (3.2),(3.4) and Eqs. (3.10),(3.12).

In summary, the resulting equations are expressed as follows:

- Update equations for U:

$$\chi^{t+1}_{(\mu i)} = \sum_r \frac{(v^t_{ir})^2}{a^t_{\mu r\to(\mu i)}+\lambda} \tag{3.27}$$

$$\Delta^{t+1}_{(\mu i)} = \sum_r u^t_{\mu r\to(\mu i)} v^t_{ir} \tag{3.28}$$

$$\hat{a}^{t+1}_{(\mu i)\to\mu r} = \frac{(v^t_{ir})^2}{1+\chi^t_{(\mu i)} - \frac{(v^t_{ir})^2}{a^t_{\mu r\to(\mu i)}+\lambda}} \tag{3.29}$$

$$\hat{b}^{t+1}_{(\mu i)\to\mu r} = \frac{y_{(\mu i)} - \Delta^t_{(\mu i)} + u^t_{\mu r\to(\mu i)} v^t_{ir}}{1+\chi^t_{(\mu i)} - \frac{(v^t_{ir})^2}{a^t_{\mu r\to(\mu i)}+\lambda}} v^t_{ir} \tag{3.30}$$

$$a^{t+1}_{\mu r} = \sum_{(\mu i)\in\partial\mu r} \hat{a}^t_{(\mu i)\to\mu r} \tag{3.31}$$

$$b^{t+1}_{\mu r} = \sum_{(\mu i)\in\partial\mu r} \hat{b}^t_{(\mu i)\to\mu r} \tag{3.32}$$

$$a^{t+1}_{\mu r\to(\mu i)} = a^t_{\mu r} - \hat{a}^t_{(\mu i)\to\mu r} \tag{3.33}$$

$$b^{t+1}_{\mu r\to(\mu i)} = b^t_{\mu r} - \hat{b}^t_{(\mu i)\to\mu r} \tag{3.34}$$

$$u^{t+1}_{\mu r\to(\mu i)} = \frac{b^t_{\mu r^\to(\mu i)}}{a^t_{\mu r\to(\mu i)}+\lambda} \tag{3.35}$$

$$u^{t+1}_{\mu r} = \frac{b^t_{\mu r}}{a^t_{\mu r}+\lambda} \tag{3.36}$$

- Update equations for V:

$$\eta^{t+1}_{(\mu i)} = \sum_r \frac{(u^{t+1}_{\mu r})^2}{c^t_{ir\to(\mu i)} + \lambda} \tag{3.37}$$

$$\Theta^{t+1}_{(\mu i)} = \sum_r v^t_{\mu r\to(\mu i)} u^{t+1}_{\mu r} \tag{3.38}$$

$$\hat{c}^{t+1}_{(\mu i)\to ir} = \frac{(u^{t+1}_{\mu r})^2}{1 + \eta^t_{(\mu i)} - \frac{(u^{t+1}_{\mu r})^2}{c^t_{ir\to(\mu i)}+\lambda}} \tag{3.39}$$

$$\hat{d}^{t+1}_{(\mu i)\to ir} = \frac{y_{(\mu i)} - \Theta^t_{(\mu i)} + v^t_{ir\to(\mu i)} u^{t+1}_{\mu r}}{1 + \eta^t_{(\mu i)} - \frac{(u^{t+1}_{\mu r})^2}{c^t_{ir\to(\mu i)}+\lambda}} u^{t+1}_{\mu r} \tag{3.40}$$

$$c^{t+1}_{ir} = \sum_{(\mu i)\in\partial ir} \hat{c}^t_{(\mu i)\to ir} \tag{3.41}$$

$$d^{t+1}_{ir} = \sum_{(\mu i)\in\partial ir} \hat{d}^t_{(\mu i)\to ir} \tag{3.42}$$

$$c^{t+1}_{ir\to(\mu i)} = c^t_{ir} - \hat{c}^t_{(\mu i)\to ir} \tag{3.43}$$

$$d^{t+1}_{ir\to(\mu i)} = d^t_{ir} - \hat{d}^t_{(\mu i)\to ir} \tag{3.44}$$

$$v^{t+1}_{ir\to(\mu i)} = \frac{d^t_{ir\to(\mu i)}}{c^t_{ir\to(\mu i)} + \lambda} \tag{3.45}$$

$$v^{t+1}_{ir} = \frac{d^t_{ir}}{c^t_{ir} + \lambda} \tag{3.46}$$

Here, t denotes the counter index for the update. It should be noted that in order to update variables for V at time t, $u^{t+1}_{\mu r}$ is used instead of $u^t_{\mu r}$. We term the algorithm composed of Eqs. (3.27)-(3.46) as cavity-based matrix factorization (CBMF).

The computational cost per update of each equation is $O(|\Omega|R)$ and the necessary memory cost corresponds to $O(|\Omega|R)$. The computational cost is competitive, and this is discussed later. Conversely, the necessary memory cost of CBMF exceeds those of ALS and SGD (Table 3.1). Although this is a disadvantage of CBMF, its necessary memory size is reduced to that of ALS and SGD by utilizing an approximation that is similar to that for deriving AMP from belief propagation [59] as shown below.

3.1.2 Derivation of the approximate algorithm

CBMF entails $O(|\Omega|R)$ memory cost, and this is equivalent to the number of edges in the factor graph. When R and c, which is the average number of the observed entries per column, are sufficiently large, the effect caused by omitting a variable node is expected to be negligible. Thus, the variables corresponding to the edges can be replaced by those corresponding to nodes. The goal of this subsection involves deriving update equations with respect to the variables corresponding to the nodes. In the following, R and c are assumed as sufficiently large.

Equation (3.29) is approximately re-expressed as follows:

$$\hat{a}_{(\mu i)\to\mu r} = \frac{v_{ir}^2}{1+\sum_s \frac{v_{is}^2}{a_{\mu s\to(\mu i)}+\lambda} - \frac{v_{ir}^2}{a_{\mu r\to(\mu i)}+\lambda}} \simeq \frac{v_{ir}^2}{1+\chi_{(\mu i)}}, \tag{3.47}$$

where $\chi_{(\mu i)}$ is also approximated by ignoring one of c terms as follows:

$$\chi_{(\mu i)} \simeq \sum_s \frac{v_{is}^2}{a_{\mu s}+\lambda}. \tag{3.48}$$

Operating $\sum_{(\mu i)\in\partial\mu r}$ on both sides of Eq. (3.47) yields

$$a_{\mu r} = \sum_{(\mu i)\in\partial\mu r} \frac{v_{ir}^2}{1+\chi_{(\mu i)}}. \tag{3.49}$$

Similarly, Eq. (3.30) is re-expressed as follows:

$$\hat{b}_{(\mu i)\to\mu r} \simeq \left(\frac{y_{(\mu i)} - \sum_s u_{\mu s\to(\mu i)} v_{is}}{1+\chi_{(\mu i)}} + \frac{v_{ir} u_{\mu r\to(\mu i)}}{1+\chi_{(\mu i)}} \right) v_{ir}, \tag{3.50}$$

where $u_{\mu s\to(\mu i)}$ is also approximated by ignoring one of R or c terms as follows:

$$u_{\mu s\to(\mu i)} \simeq u_{\mu s} - \phi_{(\mu i)} \frac{v_{is}}{a_{\mu s}+\lambda}, \tag{3.51}$$

where $\phi_{(\mu i)}$ is defined as follows:

$$\begin{aligned} \phi_{(\mu i)} &= \frac{y_{(\mu i)} - \sum_s u_{\mu s\to(\mu i)} v_{is}}{1+\chi_{(\mu i)}} && (3.52) \\ &\simeq \frac{y_{(\mu i)} - \sum_s u_{\mu s} v_{is} + \phi_{(\mu i)}\chi_{(\mu i)}}{1+\chi_{(\mu i)}}. && (3.53) \end{aligned}$$

The second line is derived from Eqs. (3.48) and (3.51). We insert Eq. (3.51) into Eq. (3.50) and operate $\sum_{(\mu i)\in\mu r}$ on both sides to yield the following expression:

$$b_{\mu r} = \sum_{(\mu i)\in\mu r} \phi_{(\mu i)} v_{ir} + u_{\mu r} \sum_{(\mu i)\in\mu r} \frac{v_{ir}^2}{1+\chi_{(\mu i)}}. \tag{3.54}$$

Similarly, the update equations (3.37)-(3.46) are re-expressed by the same procedure. Finally, the approximate update equations are summarized as follows:

- Update equations for U:

$$\chi_{(\mu i)}^{t+1} = \sum_s \frac{(v_{is}^t)^2}{a_{\mu s}^t + \lambda} \tag{3.55}$$

$$\phi_{(\mu i)}^{t+1} = \frac{y_{(\mu i)} - \sum_s u_{\mu s}^t v_{is}^t + \phi_{(\mu i)}^t \chi_{(\mu i)}^t}{1 + \chi_{(\mu i)}^t} \tag{3.56}$$

$$a_{\mu r}^{t+1} = \sum_{(\mu i) \in \partial \mu r} \frac{(v_{ir}^t)^2}{1 + \chi_{(\mu i)}^t} \tag{3.57}$$

$$b_{\mu r}^{t+1} = \sum_{(\mu i) \in \mu r} \phi_{(\mu i)}^t v_{ir}^t + u_{\mu r}^t \sum_{(\mu i) \in \mu r} \frac{(v_{ir}^t)^2}{1 + \chi_{(\mu i)}^t} \tag{3.58}$$

$$u_{\mu r}^{t+1} = \frac{b_{\mu r}^t}{a_{\mu r}^t + \lambda} \tag{3.59}$$

$$\tag{3.60}$$

- Update equations for V:

$$\eta_{(\mu i)}^{t+1} = \sum_s \frac{(u_{is}^{t+1})^2}{c_{is}^t + \lambda} \tag{3.61}$$

$$\psi_{(\mu i)}^{t+1} = \frac{y_{(\mu i)} - \sum_s u_{\mu s}^{t+1} v_{is}^t + \psi_{(\mu i)}^t \eta_{(\mu i)}^t}{1 + \eta_{(\mu i)}^t} \tag{3.62}$$

$$c_{ir}^{t+1} = \sum_{(\mu i) \in \partial ir} \frac{(u_{\mu r}^{t+1})^2}{1 + \eta_{(\mu i)}^t} \tag{3.63}$$

$$d_{ir}^{t+1} = \sum_{(\mu i) \in ir} \psi_{(\mu i)}^t u_{\mu r}^{t+1} + v_{ir}^t \sum_{(\mu i) \in ir} \frac{(u_{\mu r}^{t+1})^2}{1 + \eta_{(\mu i)}^t} \tag{3.64}$$

$$v_{ir}^{t+1} = \frac{d_{ir}^t}{c_{ir}^t + \lambda} \tag{3.65}$$

We term the algorithm composed of Eqs. (3.55)-(3.65) as the approximate cavity-based matrix factorization (ACBMF). The necessary memory cost to execute the algorithm is $O((N+M)R + |\Omega|)$, which is equivalent to the number of nodes in the factor graph. When compared to CBMF, ACBMF significantly reduces the required memory cost while the necessary computational cost is unchanged.

Additionally, one can illustrate that the fixed point of ACBMF is in agreement with that of ALS. Equation (3.53) is solved with respect to $\phi_{(\mu i)}$, and we obtain the following expression:

$$\phi_{(\mu i)} = y_{(\mu i)} - \sum_s u_{\mu s} v_{is}. \tag{3.66}$$

We insert Eqs. (3.49) and (3.66) into Eq. (3.54) to yield the following expression:

$$b_{\mu r} = \sum_{(\mu i) \in \mu r} \left(y_{(\mu i)} - \sum_s u_{\mu s} v_{is} \right) v_{ir} + u_{\mu r} a_{\mu r}. \tag{3.67}$$

From Eq. (3.26), we obtain the following expression:

$$\lambda \mathbf{u}_r = \sum_{(\mu i)\in \mu r} \left(y_{(\mu i)} - \mathbf{u}_r^T \mathbf{v}_i \right) \mathbf{v}_i. \tag{3.68}$$

We solve Eq. (3.68) with respect to $\mathbf{u}_r$ to yield the following expression:

$$\mathbf{u}_r^* = \left(\sum_{(\mu i)\in \mu r} \mathbf{v}_i \mathbf{v}_i^T + \lambda \mathbf{I}_R \right)^{-1} \left(\sum_{(\mu i)\in \mu r} y_{(\mu i)} \mathbf{v}_i \right), \tag{3.69}$$

and this is equivalent to Eq. (2.85). Similarly for $\mathbf{v}_r^*$.

In contrast to ALS, ACBMF does not completely optimize U (V) for a given V (U) in each iteration, and thus the necessary computation is reduced. Evidently, this may decrease the convergence speed. However, the complete optimization for it does not necessarily bring U (V) to a better state when V (U) is far from the convergent solution. Therefore, it is not advised to expend significant computational cost on this. Additionally, the optimization in each iteration tends to strengthen time correlations of the variables, and this may make the cavity treatment inappropriate. Actually, the results of experiments shown below indicate that this concern is the case.

3.1.3 Comparison with ALS and SGD

We briefly compare (A)CBMF with ALS and SGD. ALS and SGD are algorithms that attempt to iteratively minimize the multivariate objective function (2.85). Although their working principle is natural, the performance of these algorithms can be negatively affected by the self-feedback effect caused by cycles from the graph. Conversely, (A)CBMF reduces such effect by introducing the seemingly artificial cavity functions, and this may lead to the performance improvement. In a manner similar to ALS, (A)CBMF can also be easily parallelized, and is free from learning parameters unlike SGD.

The computational and memory costs of the four algorithms are summarized in Table 3.1. The computational cost is defined as that necessary per iteration. Given this definition, SGD only updates the variables based on the gradients although the computational cost of SGD appears the lowest. Conversely, CBMF and ALS update them with closed forms, which could offer the faster convergence than SGD in terms of the total computational time. A comparison of (A)CBMF and ALS indicates that the computational cost of the former is lower. Conversely, the memory cost of CBMF is the highest while that of ACBMF is identical to that of ALS and SGD.

3.2 Numerical Experiments

3.2.1 Synthetic Data Analysis

In order to systematically compare the performance of the four algorithms, namely ALS, SGD, and (A)CBMF (C++ implementation is available at [60]), we performed extensive

	CBMF	ACBMF	ALS	SGD
Computational costs	$O(\|\Omega\|R)$	$O(\|\Omega\|R)$	$O((\|\Omega\|R^2 + (N+M)R^3))$	$O((N+M)R)$
Memory costs	$O(\|\Omega\|R)$	$O((N+M)R + \|\Omega\|)$	$O((N+M)R + \|\Omega\|)$	$O((N+M)R + \|\Omega\|)$

Table 3.1: Comparison of computational costs to update all variables once on average. Specifically, $|\Omega|$ denotes the number of observed entries, and this is assumed to exceed or be equal to the number of variables to be determined $(N+M)R$.

numerical experiments using synthetic datasets with and without noises. A dataset for the experiment was prepared as follows: For a noiseless dataset, the original matrix Y^0 is simply provided as $Y^0 = U^0(V^0)^T$. For a noisy dataset, the original matrix $Y^0 \in \mathbb{R}^{N\times M}$ is provided from $U^0 \in \mathbb{R}^{N\times R}, V^0 \in \mathbb{R}^{M\times R}$, and $Z \in \mathbb{R}^{N\times M}$ as $Y^0 = U^0(V^0)^T + Z$, where entries of U^0 and V^0 are independently sampled from the standard Gaussian distribution while those of Z are independently and identically distributed based on a Gaussian of zero mean and variance 0.09. We randomly select "observed entries" out of Y^0 with probability of c/N where $c \sim O(1)$ denotes the average number of the observed entries per column. The collection of the observed entries constitutes the observed matrix Y. We assume that true rank R is known in advance.

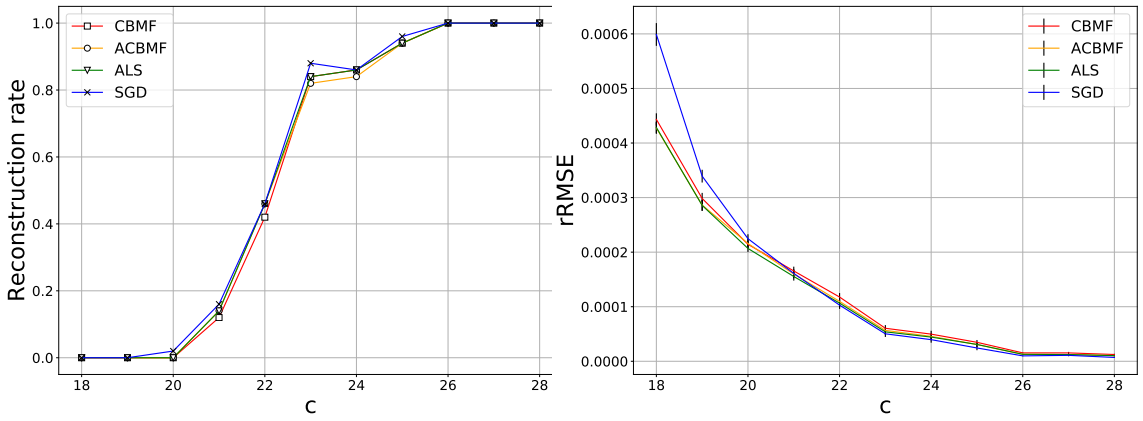
We evaluate the performance of the algorithms via the relative root mean square error (rRMSE) and the reconstruction rate. rRMSE is evaluated via the mean of the minimum value of $\sqrt{\sum_{\mu i}(y^0_{\mu i} - u_{\mu r}v_{ir})^2}/\sqrt{\sum_{\mu i}(y^0_{\mu i})^2}$ out of the ten initial conditions over 50 samples, and the recontruction rate is defined as follows: Given the effect of the regularization parameter λ and the noise Z, it is impossible to perfectly reconstruct Y^0 in the current setting. Therefore, we consider estimated factorized matrices U and V as successful if $\sqrt{\sum_{\mu i}(y^0_{\mu i} - u_{\mu r}v_{ir})^2}/\sqrt{\sum_{\mu i}(y^0_{\mu i})^2} \leq \epsilon$ holds, where ϵ is a predetermined acceptable error level. The convergence of the three algorithms is not guaranteed, and thus we attempt ten random initial conditions for each sample and algorithm and counted a "success" if at least one of the ten initial conditions leads to the successful reconstruction. In the end, the reconstruction rate is defined as the fraction of the reconstruction success over the 50 samples.

Figures 3.3 (a) and (b) plot the experimental results for the noiseless case versus the average number c of observations per column for $R = 10$, $\epsilon = 10^{-4}$ and $\lambda = 10^{-2}$. The figures show that all the algorithms exhibit similar performances. Solutions of Eq. (2.59) are characterized by Eq. (2.85), where the matrix inversion can induce numerical instability due to possible rank deficiency for too small $\lambda(> 0)$. On the other hand, as far as we investigated, the performance of the algorithms is not so sensitive as long as it is set sufficiently large. We, therefore, chose the value of λ out of several candidates ranging from 10^{-5} to 10 so that the performance differences of the algorithms are shown most significantly for both noiseless and noisy cases.

Figure 3.4a plots the results for reconstruction rate for noisy datasets. As perfect reconstruction is impossible for the noisy case, we set a larger threshold $\epsilon = 0.15$. The order in performance among the algorithms is not sensitive to the value of ϵ. The value of $\epsilon = 0.15$ is

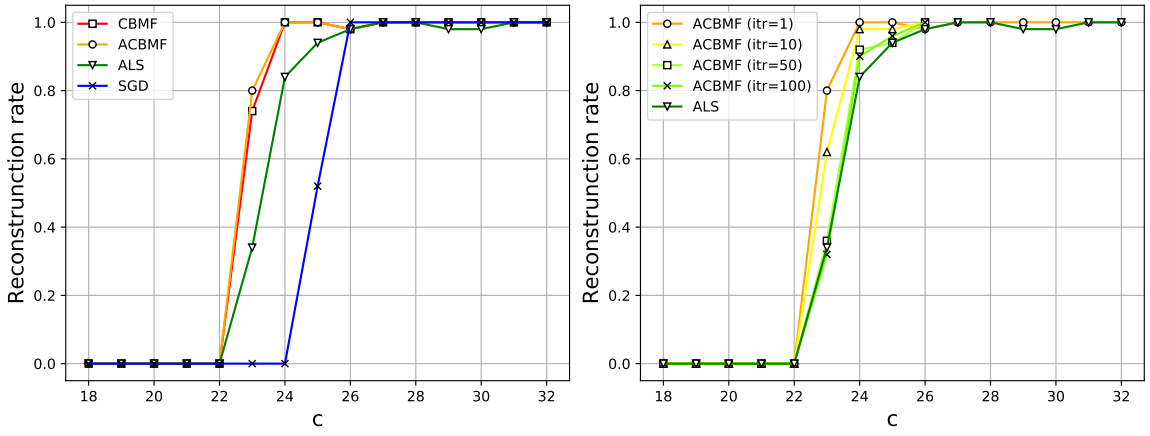
chosen out of several candidates so that the order is shown most clearly. Maximization of the reconstruction rate led to $\lambda = 10^{-2}$. The figure indicates that (A)CBMF outperforms the other algorithms. It should be noted that (A)CBMF exhibits a better reconstruction rate up to a smaller value of c than ALS while they are theoretically guaranteed to share the same fixed point. We speculate that this is because (A)CBMF weakens the self-feedback effect via the cavity treatment and by not performing optimization in each iteration. In order to verify the validity of this speculation, we examine the manner in which the reconstruction rate changes when the number of updates of ACBMF for each iteration increases, which is plotted in figure 3.4b. When the updates are repeated until convergence in each iteration, U (V) is optimized for a given V (U). This implies that the performance would become *worse* when the number of the updates increases by spending more computational cost. The figure shows that this is actually the case and supports our speculation.

Figure 3.5a shows the results for rRMSE for noisy data. The performance of SGD is significantly worse when compared to that of (A)CBMF and ALS. This is potentially because the scheduling of the learning rate used in the SGD experiments is not optimally tuned. The default scheduling that is provided in a code distribution [61] leads to a terrible result, and thus we select a better scheduling although it is non-trivial to determine the optimal one. Conversely, (A)CBMF and ALS are free from such issues as they involve no scheduling of parameters. (A)CBMF exhibits slightly better performance when compared to that ALS. Similarly, for reconstruction rate, the performance of ACBMF approaches that of ALS when the number of iterations per update increases (Fig. 3.5b).



(a) (b)

Figure 3.3: Comparison of reconstruction rate (a) and rRMSE (b) between (A)CBMF, ALS, and SGD for noiseless data as a function of c for matrices with rank $R = 10$, and system size $N = 500$, $M = 1000$. For each c, the rate was evaluated from 50 experiments where $\lambda = 10^{-2}$ was used. Reconstruction is considered as successful when $\sqrt{\sum_{\mu i}(y^0_{\mu i} - u_{\mu r}v_{ir})^2}/\sqrt{\sum_{\mu i}(y^0_{\mu i})^2} \leq 10^{-4}$ is achieved for at least once in ten trials. The figures are taken from Ref. [57].



(a) (b)

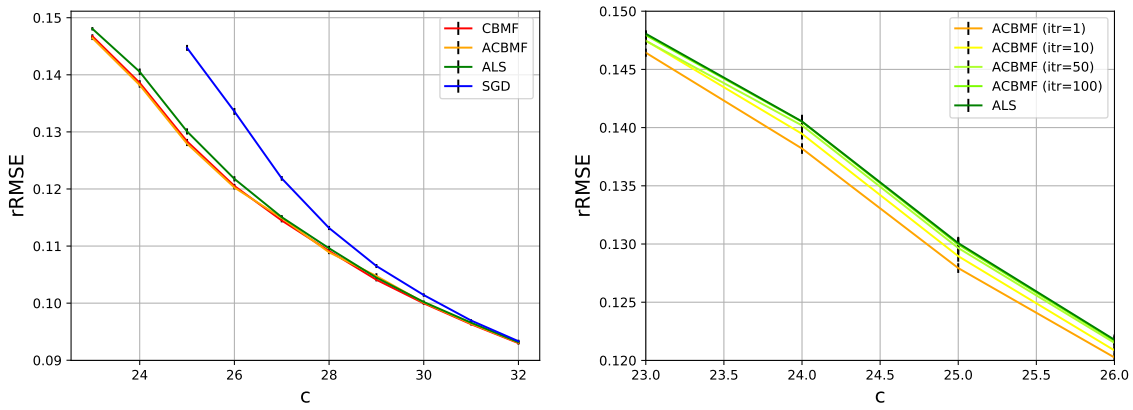
Figure 3.4: Reconstruction rate for noisy data as a function of c for matrices with rank $R = 10$, and system size $N = 500, M = 1000$. For each c, the rate was evaluated from 50 experiments where $\lambda = 10^{-2}$ was used. Reconstruction is considered as successful when $\sqrt{\sum_{\mu i}(y^0_{\mu i} - u_{\mu r}v_{ir})^2}/\sqrt{\sum_{\mu i}(y^0_{\mu i})^2} \leq 0.15$ for at least once in ten trials. (a) Comparison between (A)CBMF, ALS and SGD. (b) Results for ACMBF when the number of updates per iteration increases. The figures are taken from Ref. [57].

3.2.2 Real Data Analysis

We also examined the usefulness of the proposed algorithm via application to three benchmark datasets of recommender systems, namely MovieLens 1M, 10M, and 20M [62]. Specifically, the 1M dataset is composed of rating values s from 1 to 5 with step 1, and 10M and 20M are from 0.5 to 5 with step 0.5. The higher values correspond to higher evaluations for movies or music provided by users. Details of the datasets are summarized in Table A.1.

The performance of each algorithm for the matrix is evaluated as follows: We randomly split the matrix entries into 10 groups, matrix factorization is performed by using data of 9-of-the-10 groups, and the performance of the obtained factorization is measured by using data of the remaining group. We employ root mean square error (RMSE) as a performance measure, and it is averaged over 50 samples of the experiment. In all the experiments, we set $R = 10$. The regularization parameter λ is chosen out of several candidates so that RMSE is minimized for the test set.

Figures 3.6-3.8 show the performance measure of (A)CBMF, ALS and SGD evaluated for the three datasets. The figures represent RMSE relative to the number of iterations. The figures indicate that all the algorithms finally achieve similar performance although the number of iterations necessary for convergence is minimized for ALS. However, it should be noted that the ALS requires a significantly higher computational cost than (A)CBMF and SGD per iteration (Table3.1). Thus, (A)CBMF converges faster than the other algorithms in terms of actual time when R is relatively large.



(a) (b)

Figure 3.5: Relative root mean square error (rRMSE) of reconstructed samples as a function of c. Experimental conditions are identical to those in Figure 3.4. (a) Comparison between (A)CBMF, ALS and SGD. (b) Results for ACBMF when the number of updates per iteration increases. The figures are taken from Ref. [57].

3.3 Summary

In summary, we developed matrix factorization algorithms that are abbreviated as CBMF and ACBMF based on the cavity method. In terms of computational cost, CBMF is competitive with SGD because CBMF updates variables in closed forms (which generally reduces the number of iterations necessary for convergence) although a comparison of the necessary computational cost to update all variables once on average indicates that the computational cost of SGD is the smallest of the three. In a manner similar to CBMF, ALS updates variables in closed form although its computational cost exceeds that of CBMF because ALS requires the matrix inversion operation, which CBMF does not require. Conversely, in terms of the memory cost, CBMF requires more capacity than the others, and thus we developed ACBMF by utilizing an approximation that is similar to that for deriving AMP from belief propagation. The necessary memory cost of ACBMF is identical to that of SGD and ALS.

Experiments involving noisy synthetic data indicated that (A)CBMF exhibits better performance without the necessity of parameter tuning when observed entries are not sufficiently large. The superiority of the performance presumably stems from the reduction of self-feedback effects via the introduction of cavity treatment and avoidance of the complete optimization in each update. Experiments using real world dataset indicated that all algorithms achieved similar performance although (A)CBMF converges faster than the other two in actual time when rank R is relatively large.

Future work includes generalization of CBMF to matrix factorization problems with additional constraints such as non-negative matrix factorization [63].

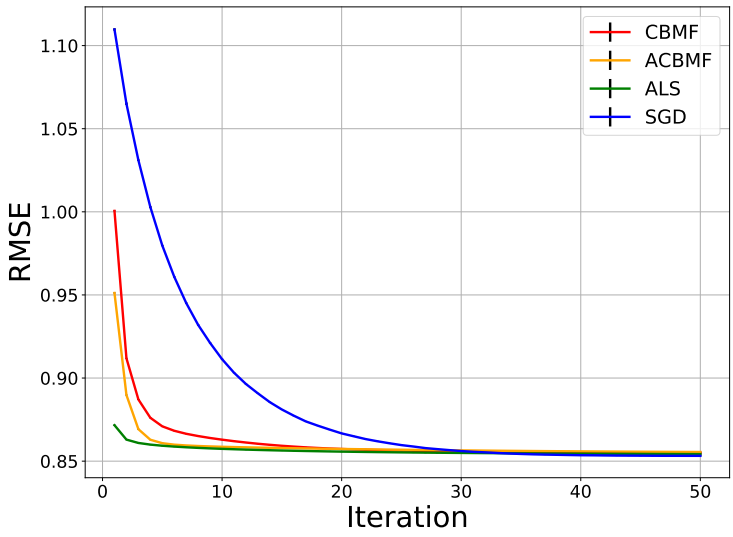


Figure 3.6: Results for MovieLens 1M dataset, RMSE is plotted versus iteration. In the experiments, we set $R = 10$, and the regularization parameter λ is fixed as 3. The figure compares the result of the four algorithms, (A)CBMF, ALS, and SGD. The figure is taken from Ref. [57].

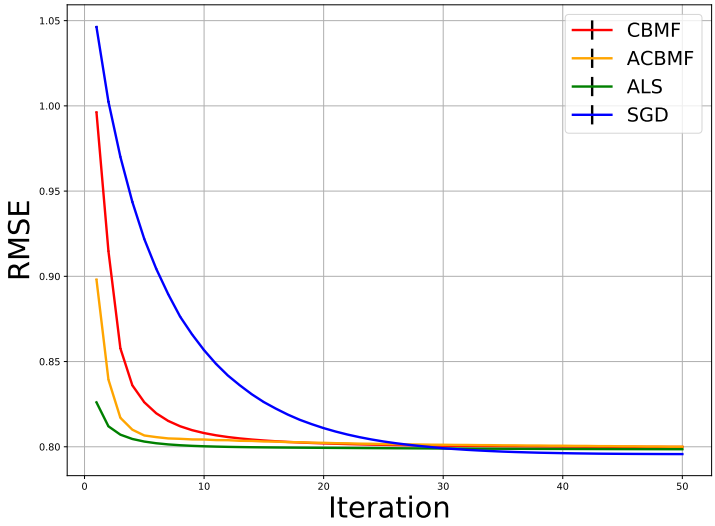


Figure 3.7: Results for MovieLens 10M dataset. Experimental conditions are identical to those in Figure 3.6. The figure is taken from Ref. [57].

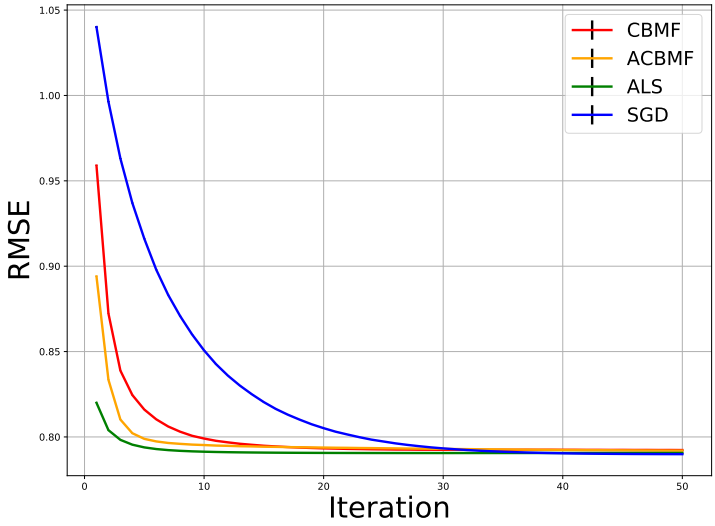


Figure 3.8: Results for MovieLens 20M dataset. Experimental conditions are identical to those in Figure 3.6. The figure is taken from Ref. [57].

Chapter 4

Fragility of spectral clustering for networks with an overlapping structure

A graph or a network that represents related data is a common data structure in multivariate statistics, machine learning, and statistical mechanics. Identifying densely connected subgraphs—community detection—is useful for graph analysis. Such subgraphs (or the corresponding node set) are referred to as communities. Spectral clustering is a popular community detection algorithm that is efficient yet highly accurate on random graph models [64, 65, 66, 67]. Nevertheless, spectral clustering often fails to identify plausible communities when it is applied to real-world networks. This is presumably because of specific features of real-world networks that are missing in simple random graph models. To fill this discrepancy, in this paper, we theoretically investigate how overlapping of communities affects accuracy of spectral clustering. We will give precise definitions of a community, an overlapping community, and the accuracy of clustering in Sec. 4.1.

We denote an undirected graph as $G = (V, E)$, where V ($|V| = N$) is a set of nodes and E ($|E| = m$) is a set of edges. The graph is represented by the $N \times N$ adjacency matrix A, where $A_{ij} = 1$ when a pair of nodes i and j is connected by an edge and $A_{ij} = 0$ otherwise. The adjacency matrix of graphs with strong (Fig. 4.1a) and weak (Fig. 4.1b) non-overlapping community structures are illustrated in Fig. 4.1.

To identify the community structure, spectral clustering [5] computes the leading eigenvalues and eigenvectors of a regularized adjacency matrix; in this paper, as an example, we focus on the so-called modularity matrix [12] as the regularized adjacency matrix. When the community structure can be clearly identified, the isolated leading eigenvectors have relevant information of the communities, while a bulk of eigenvalues emerges from the randomness of a graph. For example, Fig. 4.1c shows the spectral density of the modularity matrix corresponding to the adjacency matrix in Fig. 4.1a. In this case, the largest eigenvalue is clearly separated from the bulk of eigenvalues, and we can extract two communities using the isolated leading eigenvector. On the other hand, Fig. 4.1d shows the case corresponding to the adjacency matrix in Fig. 4.1b. The eigenvalue correlated to the community structure is buried in the bulk of eigenvalues, and the spectral density is no longer distinguishable from

that of a uniformly random graph. The phase transition point that the eigenvalues do not exhibit community structure at all is referred to as the (algorithmic) detectability limit [64, 68, 69] of spectral clustering.

As a tool for theoretical analysis, we use the replica method that originated from statistical physics. It enables us to calculate the ensemble average over random graph instances. As a result, we obtain a detectability phase diagram that indicates the effect of overlapping on spectral clustering.

Several existing studies have investigated the fragility, i.e., lack of robustness, of spectral clustering. Owing to the fact that real-world networks have more complex structures than a simple random graph, the studies have considered the fragility in case of, e.g., adversarial perturbations [70], noise perturbations [71, 72], tangles and cliques [73], and localization of eigenvectors [68, 69, 74]. In this paper, we analyze the effect of the overlapping structure on the graph spectra. Specifically, we found that, when the size of the community overlap is increased, it is the isolated eigenvalue that is mainly affected. On the other hand, it is the bulk of eigenvalues that is mainly affected when the density of the community overlap is increased.

Noted that identifying an overlapping community structure itself is not a goal of this paper. There are in fact many algorithms for such a purpose [75, 76, 77, 78, 20, 79, 80, 81, 82]. To identify or to assess an overlapping community structure, one should use a suitable algorithm. Usually, however, we do not a priori know whether communities are overlapped. Moreover, even when it is the case, it is hard to imagine that the spectral clustering becomes completely useless. Thus, we investigate how the signal of structural heterogeneity remains in the spectral clustering, although it may not be the best algorithm to use.

The rest of the paper is organized as follows. In Sec. 4.1, we introduce the overlapping random graph models that we consider. In Sec. 4.2, we provide the replica analysis for the graph spectra of the random graph model. In Sec. 4.3, we show the results and their interpretation obtained by the replica analysis. Finally, Sec. 4.4 presents a discussion.

4.1 Overlapping stochastic block model

Throughout this paper, we consider a class of random graph models called the stochastic block model (SBM). It is a random graph model that has a preassigned (planted) modular structure. Here, as a particular case of the SBM, we introduce the overlapping SBM. Although we focus only on the so-called canonical SBM in the main text, its microcanonical counterpart [21, 83] (see Appendix C for a detailed definition) is also analyzed in Appendix C.2.

4.1.1 Canonical SBM

Before considering the overlapping SBM, we first introduce the (canonical) SBM with a general structure. We define a *block* as a node set in which the nodes are statistically equivalent. A graph with K blocks is generated from the SBM as follows. For each node of the graph, we preassigned a block label $\boldsymbol{t} = [t_i]$ $t_i \in \{1, \cdots, K\}$ $(i \in V)$. Then, each pair of nodes (i, j) is connected by an edge with probability $\rho_{t_i t_j}$ independently and randomly; this

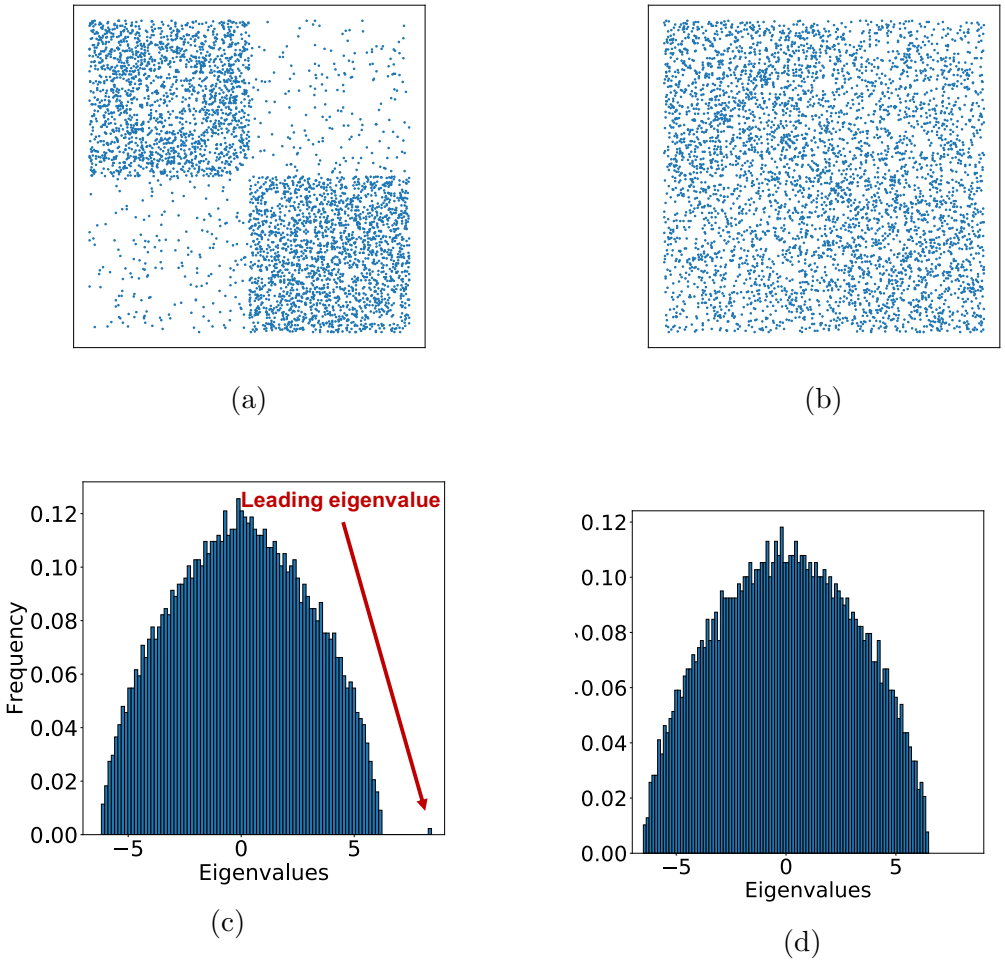


Figure 4.1: Adjacency matrices of graphs with a non-overlapping structure and the corresponding histograms of the bulk of eigenvalues of the modularity matrix. (a, c) Nodes in the same communities are more densely connected internally than externally (strong community structure). (b, d) All nodes are connected with almost the same probability (weak community structure).

probability is provided as an element of the $K \times K$ affinity matrix $\boldsymbol{\rho} = [\rho_{kl}]$, $0 \leq \rho_{kl} \leq 1$. Therefore, the probability of a graph instance is expressed as

$$P(A|K, \boldsymbol{t}, \boldsymbol{\rho}) = \prod_{i<j} \rho_{t_i t_j}^{A_{ij}} (1 - \rho_{t_i t_j})^{1 - A_{ij}}. \tag{4.1}$$

Here, because we consider undirected simple graphs, we assume that $A_{ii} = 0$ and $A_{ij} = A_{ji}$. Moreover, we focus on sparse graphs throughout this paper; i.e., we assume $\rho_{rs} = O(1/N)$ for all r and s. When every matrix element of $\boldsymbol{\rho}$ is equal, the model becomes the so-called Erdős–Rényi random graph model. We also introduce a vector that represents the block-size distribution as $\boldsymbol{p} = [p_k]$ $(k \in \{1, \ldots, K\})$, where $p_k = \sum_{i=1}^{N} \delta_{k,t_i}/N$ ($\delta_{a,b}$ represents Kronecker's delta).

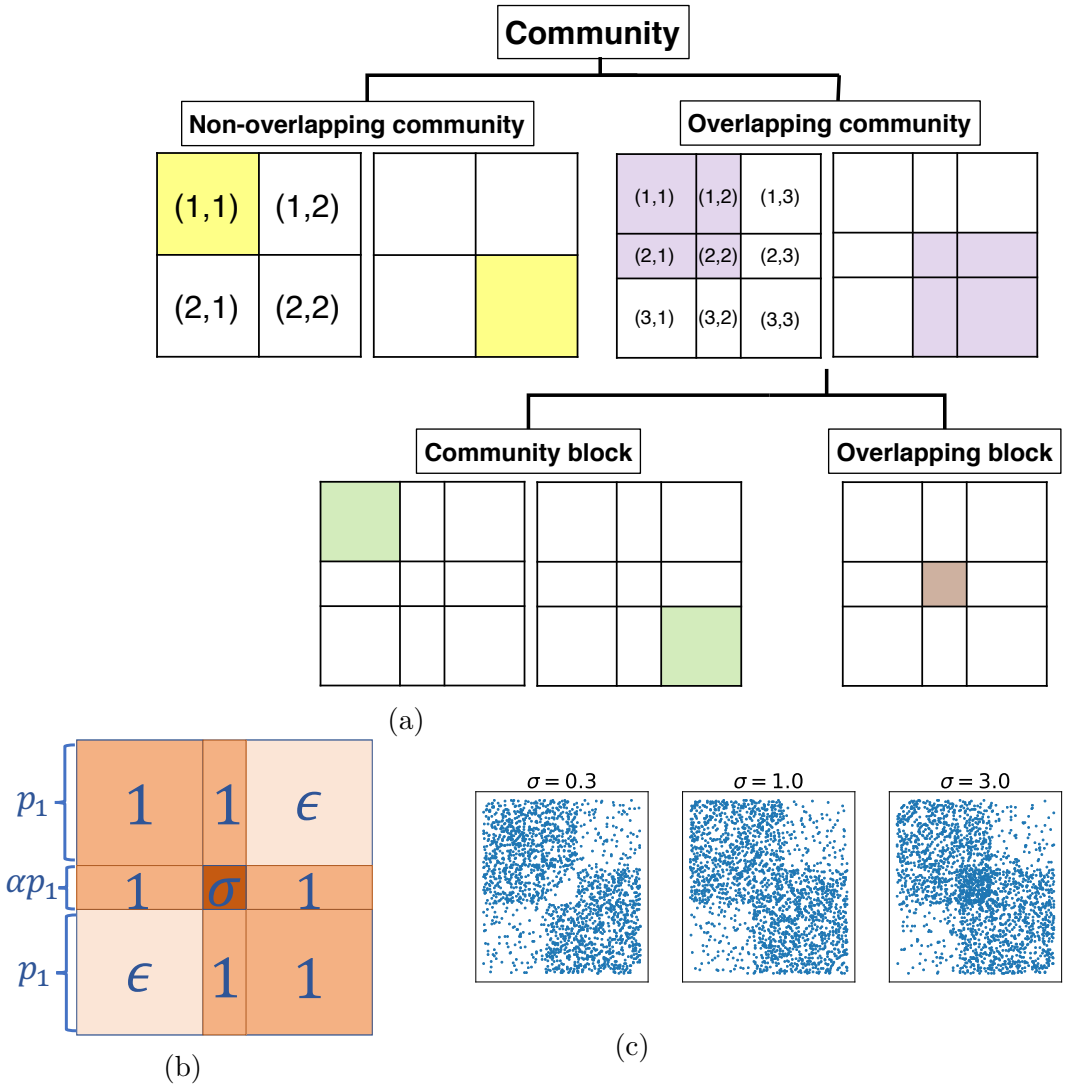


Figure 4.2: (a) Classification chart of communities and blocks using schematic pictures of adjacency matrices. Communities are classified into non-overlapping and overlapping communities, and an overlapping community consists of nodes in community blocks and nodes in an overlapping block. Each matrix element represents a pair of block labels, i.e., the corresponding set of node pairs. (b) Structure of the overlapping SBM that we consider. The node sets incident on the $(1,1)$ and $(3,3)$ elements correspond to the community blocks, while the node set incident on the $(2,2)$ element corresponds to the overlapping block. α, ϵ, and σ are the fraction of the overlapping block size, the inverse of the community structure strength, and the density of the overlapping block, respectively. The value of each block corresponds to an element of affinity matrix (4.5) divided by ρ_{in}. (c) Adjacency matrices of graph instances of the overlapping SBM with $\sigma = 0.3, 1$, and 3.

For example, a two-block SBM is parameterized as

$$\boldsymbol{p} = (p_1, p_2), \tag{4.2}$$

$$\boldsymbol{\rho} = \begin{pmatrix} \rho_{\mathrm{in}} & \rho_{\mathrm{out}} \\ \rho_{\mathrm{out}} & \rho_{\mathrm{in}} \end{pmatrix} = \begin{pmatrix} 1 & \epsilon \\ \epsilon & 1 \end{pmatrix} \rho_{\mathrm{in}}. \tag{4.3}$$

Here, edges in the $(1,1)$ and $(2,2)$ elements have the same generation probability ρ_{in}. In contrast, edges in the $(1,2)$ and $(2,1)$ elements have the generation probability ρ_{out}; $\epsilon = \rho_{\text{out}}/\rho_{\text{in}}$ is a parameter that controls the strength of the community structure. We define *non-overlapping communities* as node sets incident on the $(1,1)$ and $(2,2)$ elements, as illustrated in Fig. 4.2a.

4.1.2 Overlapping canonical SBM

We define the overlapping SBM as the three-block SBM that has parameters

$$\boldsymbol{p} = (p_1, p_2, p_3), \tag{4.4}$$

$$\boldsymbol{\rho} = \begin{pmatrix} \rho_{\text{in}} & \rho_{\text{in}} & \rho_{\text{out}} \\ \rho_{\text{in}} & \sigma\rho_{\text{in}} & \rho_{\text{in}} \\ \rho_{\text{out}} & \rho_{\text{in}} & \rho_{\text{in}} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \epsilon \\ 1 & \sigma & 1 \\ \epsilon & 1 & 1 \end{pmatrix} \rho_{\text{in}}. \tag{4.5}$$

Here, $\boldsymbol{p}$ and $\boldsymbol{\rho}$ are illustrated in Fig. 4.2b. As illustrated in Fig. 4.2a, we define the node sets incident on the sets of elements $\{(1,1), (1,2), (2,1), (2,2)\}$ and $\{(2,2), (2,3), (3,2), (3,3)\}$ as *overlapping communities*, respectively; edges therein have the same generation probability ρ_{in}, except for the $(2,2)$ element. Within the overlapping communities, we define the node sets incident on the $(1,1)$ and $(3,3)$ elements as *community blocks* and the node set incident on the $(2,2)$ element as an *overlapping block*. We let the edge generation probability of the $(1,3)$ and $(3,1)$ elements be ρ_{out} $(= \epsilon\rho_{\text{in}})$. The edge generation probability of the $(2,2)$ element is parametrized as $\sigma\rho_{\text{in}}$; σ is a parameter that controls the density of the overlapping block. Adjacency matrices with different values of σ are exemplified in Fig. 4.2c (see Appendix F for the relationship between this overlapping SBM and the mixed-membership SBM [20].)

We define the average degree of each block $\boldsymbol{c} = (c_1, c_2, c_3)$, where the degree of a node is the number of edges connected to the node. The ratio c_1/c_2 can also be expressed as $(1+\alpha+\epsilon)/(\sigma\alpha+2)$ using the affinity matrix elements, where we introduced $\alpha \equiv p_2/p_1$. Therefore, the parameters of the overlapping SBM are constrained as

$$c_1(\sigma\alpha + 2) = c_2(1 + \alpha + \epsilon). \tag{4.6}$$

For simplicity, we assume the symmetry between the community blocks, i.e., $p_1 = p_3$ and $c_1 = c_3$. We assume that the affinity matrix is symmetric, owing to the fact that we consider undirected graphs.

A technically interesting aspect of the present analysis is that this is a model-inconsistent scenario; while the overlapping SBM that we consider consists of three blocks, we consider the partitioning into two non-overlapping communities.

How to evaluate the accuracy of the spectral clustering on the overlapping SBM is an arguable issue. In this paper, we evaluate whether the community blocks are identified correctly and neglect the partitioning with respect to the overlapping block. That is, we define an accuracy of a partition as

$$\begin{aligned} &\text{Accuracy} \equiv \max\left\{ f(\hat{\boldsymbol{t}}), f\left(\mathcal{P}(\hat{\boldsymbol{t}})\right) \right\}, \\ &f(\hat{\boldsymbol{t}}) = \frac{1}{N(p_1+p_3)} \left(\sum_{i \in V_1} \delta_{\hat{t}_i,1} + \sum_{i \in V_3} \delta_{\hat{t}_i,2} \right). \end{aligned} \tag{4.7}$$

Here, $\sum_{i\in V_k}$ is the sum over the node indices belonging to the kth block, $\hat{\boldsymbol{t}} = [\hat{t}_i]$ ($\hat{t}_i \in \{1, 2\}$) is the inferred non-overlapping community label of node i. The operator $\mathcal{P}$ permutes the inferred labels; namely, $\hat{t}_i = 1$ is replaced by $\hat{t}_i = 2$ and vice versa. The maximization is required to eliminate the degrees of freedom by permutation.

4.2 Replica analysis

We now calculate the spectrum of the overlapping SBM and show that a phase transition point of the largest eigenvalue exhibits the detectability limit. It should be noted that the same result is obtained in the case of the microcanonical SBM (Appendix C.2).

4.2.1 Spectrum and the detectability limit of the overlapping SBM

As an example of a regularized adjacency matrix, we consider the modularity matrix. Each element of the matrix is defined as

$$M_{ij} = A_{ij} - \frac{d_i d_j}{2m}, \tag{4.8}$$

where d_i ($= \sum_{j=1}^{N} A_{ij}$) is the degree of a node i and m ($= |E|$) is the total number of the edges. Partitioning into two non-overlapping communities can be identified by the eigenvector of the largest eigenvalue. Thus, our goal is to solve the following maximization problem.

$$\lambda(M) = \frac{1}{N} \max_{\boldsymbol{x}} \boldsymbol{x}^\top M \boldsymbol{x}, \quad \text{subj. to } \boldsymbol{x}^\top \boldsymbol{x} = N, \tag{4.9}$$

where $\lambda(M)$ is the largest eigenvalue of M, and $\boldsymbol{x}^\top$ is the transpose of a vector $\boldsymbol{x}$. This problem can be expressed as

$$f(M, \beta) = -\frac{1}{\beta N} \log Z(M, \beta), \tag{4.10}$$

$$\lambda(M) = -2 \lim_{\beta\to\infty} f(M, \beta), \tag{4.11}$$

$$Z(M, \beta) = \int d\boldsymbol{x} e^{\frac{\beta}{2}\boldsymbol{x}^\top M \boldsymbol{x}} \delta(\boldsymbol{x}^\top \boldsymbol{x} - N), \tag{4.12}$$

where $Z(M, \beta)$ is the partition function. The constraint (4.9) is imposed by the delta function in (4.12), and taking $\beta \to \infty$ in (4.11) leads to the maximization of the exponent of the exponential function in (4.12). Because we are interested in the typical behavior of the graph instances, we analyze

$$[\lambda(M)]_M = 2 \lim_{\beta\to\infty} \frac{1}{\beta N} [\log Z(M, \beta)]_M , \tag{4.13}$$

where $[\cdots]_M$ represents the ensemble average over graph instances. Unfortunately, it is difficult to calculate the average $[\log Z(M, \beta)]_M$ analytically. To overcome this difficulty, we use the replica trick, namely,

$$[\log Z(M, \beta)]_M = \lim_{n\to 0} \frac{\partial}{\partial n} \log[Z^n(M, \beta)]_M. \tag{4.14}$$

Here, the exponent n in $[Z^n]_M$ is a real value. However, we treat n as an integer for a moment. In the end, we perform the analytic continuation to the real value. This treatment is termed the replica method.

From Eq. (4.12), the nth moment the partition function is obtained as

$$[Z^n(M,\beta)]_M = \int \left(\prod_{a=1}^{n} d\boldsymbol{x}_a \delta(\boldsymbol{x}_a^\top \boldsymbol{x}_a - N)\right) \left[\exp\left(\frac{\beta}{2}\sum_a \boldsymbol{x}_a^\top M \boldsymbol{x}_a\right)\right]_M , \tag{4.15}$$

where $a \in \{1,\dots,n\}$ is an index of n identical copies. For further calculations, we introduce several order parameters and approximations. Detailed calculations are described in Appendix B. As a result, the average largest eigenvalue in the limit of $N \to \infty$ is obtained by the following saddle-point (extremum) condition of nine auxiliary variables $(\phi, \Omega, \hat{\Omega}, m_{1k}, m_{2k}, \hat{m}_{1k}, \hat{m}_{2k}, a_k, \hat{a}_k)$.

$$\begin{aligned}
[\lambda(M)]_M = \underset{\phi,\Omega,\hat{\Omega},m_{1k},m_{2k},\hat{m}_{1k},\hat{m}_{2k},a_k,\hat{a}_k}{\mathrm{extr}} \Bigg\{ & \phi + 2\hat{\Omega}\Omega - \Omega^2 \\
& + \frac{1}{2}N\sum_{k,k'} W_{kk'} \left(\frac{a_{k'}\left(m_{2k} - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}} + \frac{4\hat{\Omega}^2}{\bar{c}}\right) + 2m_{1k'}\left(m_{1k} - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}}\right) + a_k m_{2k'}}{a_k a_{k'} - 1} \right. \\
& \left. - \frac{m_{2k} - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}} m_{1k} + \frac{4\hat{\Omega}^2}{c}}{a_k} - \frac{m_{2k'}}{a_{k'}} \right) \\
& - \sum_k p_k c_k \left(\frac{m_{2k} + 2m_{1k}\hat{m}_{1k} + \hat{m}_{2k}}{a_k - \hat{a}_k} - \frac{m_{2k'}}{a_{k'}} \right) \\
& + \frac{1}{N}\sum_k \sum_{i\in V_k} \sum_{d=0}^{\infty} \frac{\mathcal{P}_{c_k}(d)}{\phi - d\hat{a}_k} \left(d\hat{m}_{2k} + d(d-1)\hat{m}_{1k}^2 \right) \Bigg\} .
\end{aligned} \tag{4.16}$$

Here, W_{kl} and $\bar{c}$ are defined as $W_{kl} \equiv p_k \rho_{kl} p_l$ and $\bar{c} \equiv 2m/N$, respectively. $\mathcal{P}_{c_k}(d)$ is the Poisson probability mass function of degree d of each node in block k that has expectation c_k. m_{1k} is the the mean of the largest eigenvector elements that corresponds to the kth block. m_{1k} plays an important role in the derivation of the detectability limit. Definitions and interpretations of the other auxiliary variables are omitted here, because they are not directly relevant to the detectability limit (see Appendix B for the precise definitions).

The detectability limit is derived by solving the equations of the nine auxiliary variables. In particular, m_{11} $(= -m_{13})$ plays an important role for the detectability limit. When $m_{11}^2 > 0$, the spectral clustering retains the ability to detect the community structure better than a random guess (detectable condition). On the other hand, when $m_{11}^2 = 0$, the result of spectral clustering is uncorrelated to the planted structure (undetectable condition). Accordingly, the phase transition point is derived by the condition $m_{11}^2 = 0$. This corresponds to the condition that the largest eigenvalue is buried in the bulk of the eigenvalues, as we mentioned in Introduction.

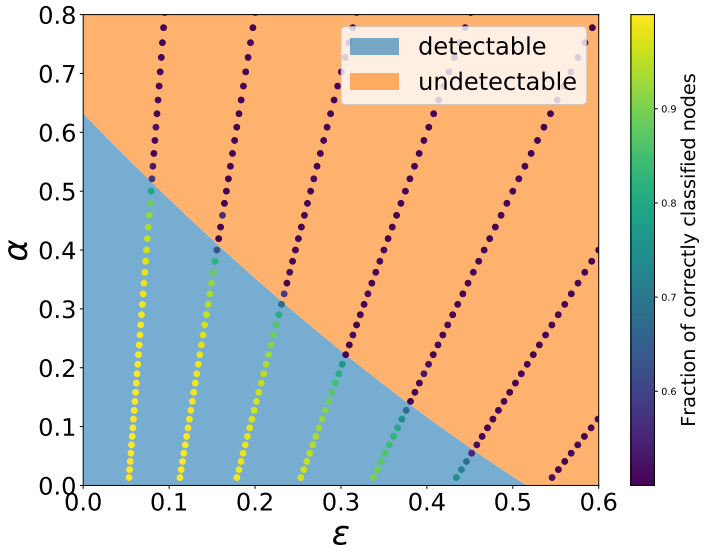


Figure 4.3: Detectability phase diagram of the (ϵ, α) plane. The model parameters are set to $c_1 = 10$ and $\sigma = 2$. Along the line of the results of the numerical experiments determined by constraint (4.6), degree c_2 takes a fixed value. The lines in this figure, from left to right, correspond to the values of c_2 from 19 to 11.

4.3 Accuracy of the spectral clustering on the overlapping SBM

In this section, using the results obtained by the replica analysis, we show how the size and density of the overlapping block affect the spectrum. We also check the validity of our analytical calculations by comparing them to the results of numerical experiments. Here, we use the microcanonical SBM in the numerical experiments instead of the canonical SBM. Here, for a technical reason that we describe in Appendix C.4, we use the microcanonical counterpart of the SBM. We used graph-tool [84] to generate graph instances of the microcanonical SBM.

4.3.1 Detectability phase diagram and the leading eigenvalue

First, to observe the overall dependency of overlapping structures, we show the detectability phase diagram. Figure 4.3 shows the detectability phase diagram of the (ϵ, α) plane. As mentioned above, ϵ is the parameter that controls the strength of the community structure and $\alpha = p_2/p_1$ is the ratio of the overlapping block and a community block. The boundary between the blue and orange regions represents the detectability limit of the spectral clustering predicted by the replica analysis. The dots represent the results of the numerical experiments; the color gradient represents the accuracy defined in Eq. (4.7). We can see that both boundaries are in a good agreement. Note that the numerical experiment is possible

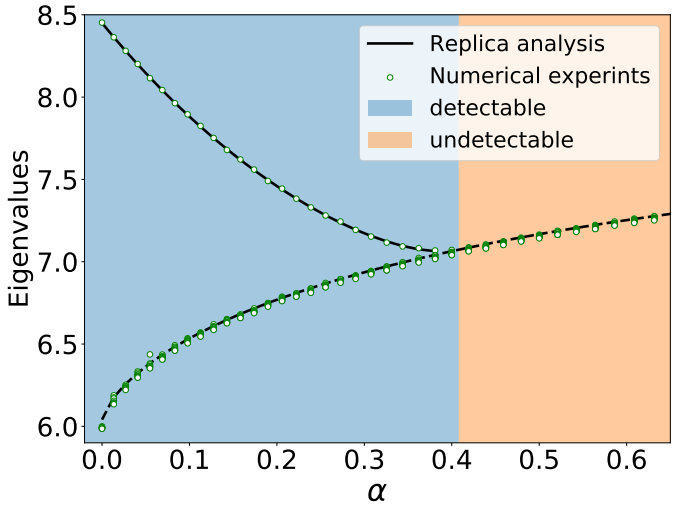


Figure 4.4: Eigenvalues derived by the replica analysis as a function of α. We set $c_1 = 10$, $c_2 = 18$, and $\sigma = 2$. The solid and dashed lines represent the isolated leading eigenvalues and the bulk edges of eigenvalues, respectively. The boundary between the blue and orange regions represents the detectability limit. The green dots represent the top ten eigenvalues computed in the numerical experiments.

only on specific curves in the parameter space because of constraint (4.6), and c_2 can take only natural numbers in the microcanonical SBM. In this experiment, we set $c_1 = 10$ and $\sigma = 2$. Then, the range c_2 can take is restricted between 11 and 19 because of the assortative condition $0 \leq \epsilon \leq 1$. This phase diagram is the result that shows how fragile the spectral clustering is against the overlapping structure.

Figure 4.4 shows the leading eigenvalue and the edge of the bulk of the eigenvalues[1], which are predicted by the replica analysis, and the top ten eigenvalues computed in the numerical experiments. We can confirm that the replica analysis accurately describes the behavior of numerical experiments. When α is small, the leading eigenvalue is separated from the bulk of the eigenvalues. As α increases, the leading eigenvalue approaches the bulk of the eigenvalues. As we described in Introduction, when it reaches the bulk of the eigenvalues, the spectral clustering loses ability to detect the community structure, i.e., the detectability limit. Note the value of ϵ also varies according to (4.6) as α varies. Thus, the horizontal axis in Fig. 4.4 corresponds to the line in Fig. 4.3 with $c_2 = 18$.

4.3.2 Effects of the size of the overlapping structure

We now investigate the effect of the overlapping structure on the accuracy of the spectral clustering when we increase the size of the overlapping block. Because the overlapping block can have denser (or sparser) edge density than the other blocks, the average degree also increases (or decreases) accordingly, as the size of the overlapping block increases. This

[1]The edge of the bulk of eigenvalue is derived as the largest eigenvalue under the undetectable condition.

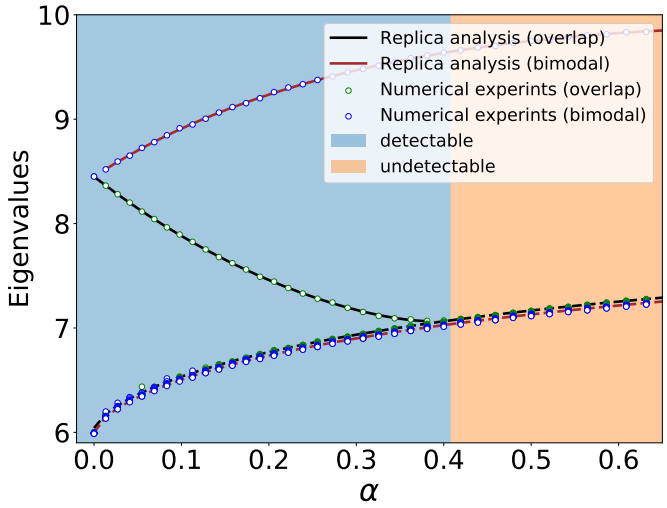


Figure 4.5: Comparison between the overlapping and bimodal SBMs. This figure shows the eigenvalues of bimodal SBM in addition to those in Fig. 4.4. The blue dots represent the top ten eigenvalues of the bimodal SBM computed in the numerical experiments. The brown solid and dashed lines represent the leading eigenvalue and the bulk edge of the eigenvalues of the bimodal SBM that are derived by the replica analysis, respectively. The spectrum of the overlapping SBM is plotted as in Fig. 4.4. These models are identical only when $\alpha = 0$. However, their bulk edges should coincide when $\alpha = 0$ and $\alpha = 1$. For the value of ϵ of the bimodal SBM, we used the same value as the overlapping SBM, which varies as α increases owing to constraint (4.6).

implies that the width of the bulk of the eigenvalues is trivially influenced, because the bulk is known to depend on the average degree [64].

However, it is not trivial if it is the only effect. Namely, the overlapping structure may affect the isolated eigenvalue or the bulk in another way. To assess the effect of the overlapping structure rather than the effect of the average degree, we compare the overlapping SBM with the model with no overlapping structures but that has the same degree distribution as the overlapping SBM. In the case of the microcanonical overlapping SBM, the degree distribution is bimodal: all the nodes in the overlapping block have the same degree, while all the other nodes have the other degree. Therefore, we consider the non-overlapping SBM with a bimodal degree distribution (see Appendix D for a detailed definition). We assume that the sizes of the blocks are equal. Hereafter, we refer to this model as the bimodal SBM.

Figure 4.5 shows the bulks of eigenvalues and the leading eigenvalues of the overlapping and bimodal SBMs. We can confirm that both bulk edges almost coincide. In contrast, the leading eigenvalue of the bimodal SBM is separated from the bulk in the whole space, while that of the overlapping SBM approaches to its bulk as α increases. This indicates that the increase of the size of the overlapping block mainly affects the leading eigenvalue instead of the bulk.

The fact that the bulk is not considerably affected is not very trivial. If we take a closer

look, the bulk edges do not exactly coincide in Fig. 4.5, although the deviation is very small. This is because the models are not identical even when there is no community structure (i.e., $\epsilon = 1$). When $\alpha = 0$, the two models reduce to the c_1-regular SBM. Thereby, their bulk edges become equal to $2\sqrt{c_1 - 1}$. When $\alpha = 1$, the overlapping SBM becomes a uniform (one block) model with (average) degree c_2, while the bimodal SBM has the community structure with (average) degree c_2. However, the bulk edge of the SBM with no overlapping blocks depends only on its average degree. Thus, although the models are not identical, their bulk edges are both $2\sqrt{c_2 - 1}$.

4.3.3 Effects of the density of the overlapping structure

Next, we investigate how density σ of the overlapping block affects the detectability. As mentioned in the previous subsection, the higher density of the overlapping block trivially makes the width of the bulk of the eigenvalues expand wider.

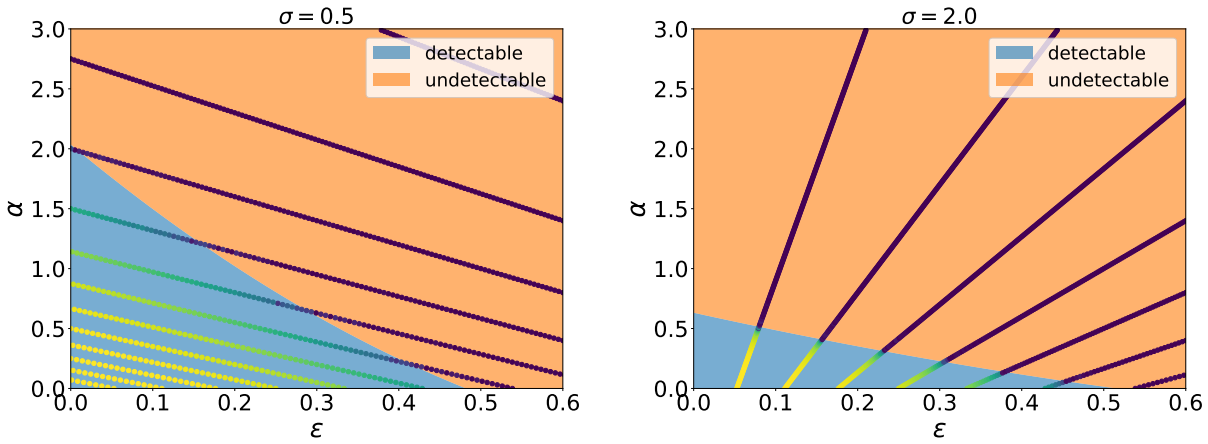
Figures 4.6a–4.6b show the detectability phase diagram derived by the replica analysis and the results of the corresponding numerical experiments for $\sigma = 0.5$ and 2. Notably, the detectable region is wider when σ is small. This indicates that the higher density deteriorates the detectability more significantly.

Let us examine σ dependency. Figure 4.7a shows the α dependencies derived by the replica analysis of the canonical SBM. They are the isolated leading largest eigenvalues and the bulk of the eigenvalues for $\sigma = 0, 0.5, 1, 1.5$, and 2. Interestingly, the isolated largest eigenvalue does not depend on σ considerably. In contrast, the bulk is highly dependent on σ. This indicates that the deterioration of the detectability due to σ is caused by the expansion of the bulk rather than the shrinkage of the isolated leading eigenvalue. Figure 4.7b similarly shows the ϵ dependencies. Again, we can see that the isolated largest eigenvalue does not depend on σ considerably while the bulk is highly dependent.

Notably, we cannot test the result of Fig. 4.7a directly in numerical experiments, because α cannot be varied continuously as ϵ is fixed. This is due to the constraints of the microcanonical SBM. Similarly, in Fig. 4.7b, ϵ cannot be varied continuously as α is fixed. Nevertheless, we can draw smooth curves in the replica analysis, because we consider the canonical SBM that is not subject to the constraints of the microcanonical SBM. Importantly, the results of the microcanonical SBM coincide with those of the canonical SBM with the regular approximation at the points where the microcanonical SBM is realizable. We also note that (Appendix C.2) the distinction between the canonical and microcanonical SBMs is invisible in infinite graph size limits.

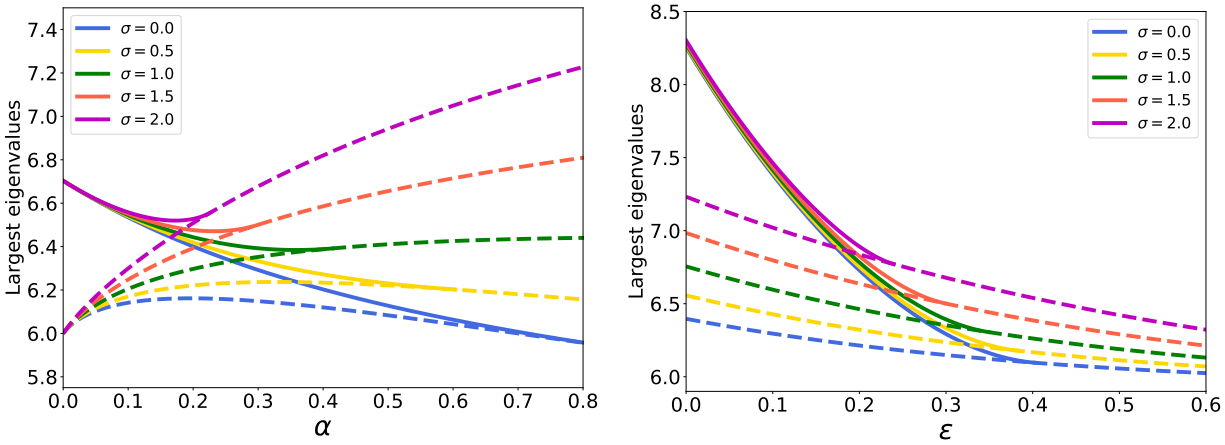
4.4 Summary

We investigated the effect of the size and the density of the overlapping block on the accuracy of spectral clustering using the replica method. Both larger size and higher density help the isolated eigenvalue to be buried in the bulk of the eigenvalues, i.e., deteriorate the detectability. Importantly, however, their mechanisms are strikingly different. We found that increasing the size of the overlapping block has a prominent effect on making the iso-



(a) (b)

Figure 4.6: Detectability phase diagram of the (ϵ, α) plane for $\sigma = 0.5,\ 2$ and $c_1 = 10$. A detailed explanation is provided in the caption of Fig. 4.3.



(a) (b)

Figure 4.7: (a) Isolated eigenvalues (solid lines) and bulk edges (dashed lines) as a function of α for $\sigma = 0, 0.5, 1, 1.5, 2$. Parameters are set to $c_1 = 10$ and $\epsilon = 0.3$. The value of degree c_2 varies according to (4.6). (b) Isolated eigenvalues (solid lines) and bulk edges (dashed lines) as a function of ϵ. α is fixed as 0.3. Other experimental conditions are identical to those of Fig. 4.7a.

lated eigenvalue smaller (Fig. 4.5). In contrast, increasing the density of the overlapping block makes the bulk width larger, while the isolated eigenvalue remains almost the same (Fig. 4.7a).

According to our findings, the results of the replica analysis are consistent with those of the numerical experiments. This indicates that the detectability phase transition of the spectral clustering in the present setting is regarded as a phenomenon that can be understood

in the scope of the mean-field theory.

Although spectral clustering typically deals with non-overlapping structures, we showed that it is also possible to analyze the model-inconsistent case, such as the overlapping SBM. It is possible, in principle, to investigate even more complex situations using the replica method. However, for example, we would need to deal with saddle-point equations with many variables if we were to analyze a general three-block SBM. Therefore, we believe that the present model is an extreme case where the analytical calculation is executable and the results are interpretable.

Chapter 5

Conclusion

5.1 Summary of the achievements

This thesis studied sparse low-rank matrix factorization problems employing notions and techniques of statistical mechanics. Such types of matrix factorization are often used in analyzing real-world data. However, the real-world data is often difficult to deal with, from which we are required to address many issues that arise. With this in mind, we studies the problems from the following two perspective.

First, we developed efficient algorithms of matrix factorization, which enables us to deal with significantly large data. This perspective is practically important because the real-world data is often given with significantly large sizes. We proposed two approximate algorithms, which are termed CBMF and ACBMF, for matrix factorization. The proposed algorithms are performable with low computational costs in parallel and distributed manners, which is preferred for practical use. Their usefulness is tested by application to matrix completion problems. By numerical experiments, we verified that our algorithms outperform existing major algorithms, ALS and SGD. This outperformance presumably originates from avoiding extra computation and weakening the self-feedback effect via the cavity treatment. Compared to the existing algorithms in terms of computational cost, our algorithms are superior to ALS, while they are competitive with SGD. However, it is known that the SGD is highly sensitive to the learning rate in its update equations and required its delicate scheduling. By contrast, our algorithms allow us to avoid such troublesome.

Second, we theoretically analyzed possibilities and limitations of matrix factorization. We tool up the spectral clustering, which is a popular method for community detection. Overlapping structures are often contained in real-world networks and generally deteriorate the performance of the spectral clustering. We examined this deterioration mechanism employing the replica method. More precisely, we assessed the algorithm accuracy when varying the model parameters, which are corresponding to the size and density of the overlapping structures, of the overlapping SBM. From this, it was revealed that the deterioration mechanism was different depending on which model parameters are varied. Increasing the size of the overlapping structure makes the isolated eigenvalue smaller, while increasing the density of the overlapping structure makes the width of the bulk of eigenvalues larger. In either case, eventually the spectral clustering loses the ability to detect community structures as

the isolated eigenvalue is no longer distinguishable from the bulk of eigenvalues.

5.2 Future study

The usefulness of the developed algorithms, CBMF and ACBMF, was demonstrated for matrix completion problems. However, it can be applicable to various matrix factorization problems. For instance, they are also applicable to the community detection. Although the factorized matrices generated from our algorithms do not have the orthogonal condition unlike the spectral clustering, they can still be utilized for finding communities in conjunction with the use of k-means clustering. This line of methods are called the graph factorization [85]. Our algorithm may improve its performance compared to the standard graph factorization algorithm, as well as the matrix completion. Besides, the proposed algorithms can be extended for performing the non-negative matrix factorization (NMF). This extension is relatively easy because we are just required to additionally impose the non-negative constraint on them. This addition can be done naively in the update rules of the proposed methods. Due to the non-negative constraint, the interpretability of the factorized matrices can be higher because they are directly viewed as a probability distribution or group assignment of belonging to each group.

Our theoretical analysis of the spectral clustering revealed that the density of the overlapping structure considerably affects the bulk of the eigenvalues, but not the isolated eigenvalue. This result is particularly interesting. It can provide rooms for application to practical algorithms such as detection of overlapping structures. Although our study focused on the overlapping structure, real-world networks often contain various kinds of structures such as link labels and hierarchical structures, as well as the overlapping structures. Investigating the influence of such structures on the spectral clustering performance is left for future work. In addition, investigating how the addition of the non-negative constraint affects its performance is of interest. This modification allows us to expect to improve the performance when a graph contains overlapping structures. Finally, the difference in how to deal with the sparse matrix between the matrix completion and community detection is also of interest. In the community detection, most of its entries are regarded as zero, whereas in the matrix completion, most of them are unobserved. In the latter case, regarding variables as unobserved allows them to have rooms to take non-zero values. Investigating how this difference influences their performance is also left for future study.

Bibliography

[1] John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[2] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[3] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. "Deepwalk: Online learning of social representations". In: *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2014, pp. 701–710.

[4] Carl Doersch. "Tutorial on variational autoencoders". In: *arXiv preprint arXiv:1606.05908* (2016).

[5] Ulrike Von Luxburg. "A tutorial on spectral clustering". In: *Statistics and computing* 17.4 (2007), pp. 395–416.

[6] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[7] Helmut Lütkepohl. *Handbook of matrices*. Vol. 1. Wiley Chichester, 1996.

[8] Rasmus Munk Larsen. "Lanczos bidiagonalization with partial reorthogonalization". In: *DAIMI Report Series* 537 (1998).

[9] Daniel D Lee and H Sebastian Seung. "Algorithms for non-negative matrix factorization". In: *Advances in neural information processing systems*. 2001, pp. 556–562.

[10] Mark EJ Newman. "Spectral methods for community detection and graph partitioning". In: *Physical Review E* 88.4 (2013), p. 042822.

[11] Mark EJ Newman. "Equivalence between modularity optimization and maximum likelihood methods for community detection". In: *Physical Review E* 94.5 (2016), p. 052315.

[12] Mark E J Newman. "Modularity and community structure in networks". In: *Proceedings of the national academy of sciences* 103.23 (2006), pp. 8577–8582.

[13] Boris V Cherkassky and Andrew V Goldberg. "On implementing the push—relabel method for the maximum flow problem". In: *Algorithmica* 19.4 (1997), pp. 390–410.

[14] 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.

[15] 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.

[16] Mark EJ Newman. "Assortative mixing in networks". In: *Physical review letters* 89.20 (2002), p. 208701.

[17] Xiao Zhang, Travis Martin, and Mark EJ Newman. "Identification of core-periphery structure in networks". In: *Physical Review E* 91.3 (2015), p. 032803.

[18] Simon Heimlicher, Marc Lelarge, and Laurent Massoulié. "Community detection in the labelled stochastic block model". In: *arXiv preprint arXiv:1209.2910* (2012).

[19] Brian Karrer and Mark EJ Newman. "Stochastic blockmodels and community structure in networks". In: *Physical review E* 83.1 (2011), p. 016107.

[20] Edoardo M Airoldi et al. "Mixed membership stochastic blockmodels". In: *Journal of machine learning research* 9.Sep (2008), pp. 1981–2014.

[21] Tiago P Peixoto. "Bayesian stochastic blockmodeling". In: *arXiv preprint arXiv:1705.10225* (2017).

[22] Emmanuel Abbe. "Community detection and stochastic block models: recent developments". In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 6446–6531.

[23] Mark EJ Newman. "Community detection in networks: Modularity optimization and maximum likelihood are equivalent". In: *arXiv preprint arXiv:1606.02319* (2016).

[24] J-J Daudin, Franck Picard, and Stéphane Robin. "A mixture model for random graphs". In: *Statistics and computing* 18.2 (2008), pp. 173–183.

[25] Aurelien Decelle et al. "Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications". In: *Physical Review E* 84.6 (2011), p. 066106.

[26] Mark EJ Newman and Gesine Reinert. "Estimating the number of communities in a network". In: *Physical review letters* 117.7 (2016), p. 078301.

[27] Andrea Lancichinetti, Santo Fortunato, and Filippo Radicchi. "Benchmark graphs for testing community detection algorithms". In: *Physical review E* 78.4 (2008), p. 046110.

[28] Patrick O Perry and Patrick J Wolfe. "Null models for network data". In: *arXiv preprint arXiv:1201.5871* (2012).

[29] Yehuda Koren, Robert Bell, and Chris Volinsky. "Matrix factorization techniques for recommender systems". In: *Computer* 8 (2009), pp. 30–37.

[30] Maryam Fazel. "Matrix rank minimization with applications". PhD thesis. PhD thesis, Stanford University, 2002.

[31] Emmanuel J Candès and Benjamin Recht. "Exact matrix completion via convex optimization". In: *Foundations of Computational mathematics* 9.6 (2009), p. 717.

[32] Emmanuel J Candès and Terence Tao. "The power of convex relaxation: Near-optimal matrix completion". In: *IEEE Transactions on Information Theory* 56.5 (2010), pp. 2053–2080.

[33] Benjamin Recht. "A simpler approach to matrix completion". In: *Journal of Machine Learning Research* 12.Dec (2011), pp. 3413–3430.

[34] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. "Matrix completion from a few entries". In: *IEEE transactions on information theory* 56.6 (2010), pp. 2980–2998.

[35] Rong Ge, Jason D Lee, and Tengyu Ma. "Matrix completion has no spurious local minimum". In: *Advances in Neural Information Processing Systems*. 2016, pp. 2973–2981.

[36] James Bennett, Stan Lanning, et al. "The netflix prize". In: *Proceedings of KDD cup and workshop*. Vol. 2007. Citeseer. 2007, p. 35.

[37] Emmanuel J Candes and Yaniv Plan. "Matrix completion with noise". In: *Proceedings of the IEEE* 98.6 (2010), pp. 925–936.

[38] Sheldon M Ross et al. *A first course in probability*. Vol. 7. Pearson Prentice Hall Upper Saddle River, NJ, 2006.

[39] David Gross. "Recovering low-rank matrices from few coefficients in any basis". In: *IEEE Transactions on Information Theory* 57.3 (2011), pp. 1548–1566.

[40] Lieven Vandenberghe and Stephen Boyd. "Semidefinite programming". In: *SIAM review* 38.1 (1996), pp. 49–95.

[41] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. "A singular value thresholding algorithm for matrix completion". In: *SIAM Journal on optimization* 20.4 (2010), pp. 1956–1982.

[42] Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. "Spectral regularization algorithms for learning large incomplete matrices". In: *Journal of machine learning research* 11.Aug (2010), pp. 2287–2322.

[43] Nathan Srebro, Jason Rennie, and Tommi S Jaakkola. "Maximum-margin matrix factorization". In: *Advances in neural information processing systems*. 2005, pp. 1329–1336.

[44] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization". In: *SIAM review* 52.3 (2010), pp. 471–501.

[45] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. "Low-rank matrix completion using alternating minimization". In: *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM. 2013, pp. 665–674.

[46] Moritz Hardt. "Understanding alternating minimization for matrix completion". In: *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*. IEEE. 2014, pp. 651–660.

[47] Moritz Hardt and Mary Wootters. "Fast matrix completion without the condition number". In: *Conference on learning theory*. 2014, pp. 638–678.

[48] Trevor Hastie et al. “Matrix completion and low-rank SVD via fast alternating least squares”. In: *The Journal of Machine Learning Research* 16.1 (2015), pp. 3367–3402.

[49] Benjamin Recht and Christopher Ré. “Parallel stochastic gradient algorithms for large-scale matrix completion”. In: *Mathematical Programming Computation* 5.2 (2013), pp. 201–226.

[50] Yunhong Zhou et al. “Large-scale parallel collaborative filtering for the netflix prize”. In: *International Conference on Algorithmic Applications in Management*. Springer. 2008, pp. 337–348.

[51] Hsiang-Fu Yu et al. “Scalable coordinate descent approaches to parallel matrix factorization for recommender systems”. In: *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE. 2012, pp. 765–774.

[52] Yong Zhuang et al. “A fast parallel SGD for matrix factorization in shared memory systems”. In: *Proceedings of the 7th ACM conference on Recommender systems*. ACM. 2013, pp. 249–256.

[53] Jason T Parker, Philip Schniter, and Volkan Cevher. “Bilinear generalized approximate message passing—Part I: Derivation”. In: *IEEE Transactions on Signal Processing* 62.22 (2014), pp. 5839–5853.

[54] Jason T Parker, Philip Schniter, and Volkan Cevher. “Bilinear generalized approximate message passing—Part II: Applications”. In: *IEEE Transactions on Signal Processing* 62.22 (2014), pp. 5854–5867.

[55] Yoshiyuki Kabashima et al. “Phase transitions and sample complexity in bayes-optimal matrix factorization”. In: *IEEE Transactions on Information Theory* 62.7 (2016), pp. 4228–4265.

[56] Ryosuke Matsushita and Toshiyuki Tanaka. “Low-rank matrix reconstruction and clustering via approximate message passing”. In: *Advances in Neural Information Processing Systems*. 2013, pp. 917–925.

[57] Chihiro Noguchi and Yoshiyuki Kabashima. “Approximate matrix completion based on cavity method”. In: *Journal of Physics A: Mathematical and Theoretical* 52.42 (2019), p. 424004.

[58] Marc Mezard and Andrea Montanari. *Information, physics, and computation*. Oxford University Press, 2009.

[59] Yoshiyuki Kabashima. “A CDMA multiuser detection algorithm on the basis of belief propagation”. In: *Journal of Physics A: Mathematical and General* 36.43 (2003), p. 11111.

[60] https://github.com/chnoguchi/cbmf.

[61] http://i.stanford.edu/hazy/hazy/victor/jellyfish/.

[62] F Maxwell Harper and Joseph A Konstan. “The movielens datasets: History and context”. In: *Acm transactions on interactive intelligent systems (tiis)* 5.4 (2016), p. 19.

[63] Daniel D Lee and H Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755 (1999), p. 788.

[64] Raj Rao Nadakuditi and Mark E J Newman. "Graph spectra and the detectability of community structure in networks". In: *Physical review letters* 108.18 (2012), p. 188701.

[65] Florent Krzakala et al. "Spectral redemption in clustering sparse networks". In: *Proceedings of the National Academy of Sciences* 110.52 (2013), pp. 20935–20940.

[66] Elchanan Mossel, Joe Neeman, and Allan Sly. "Belief propagation, robust reconstruction and optimal recovery of block models". In: *Conference on Learning Theory*. 2014, pp. 356–370.

[67] Emmanuel Abbe et al. "Entrywise eigenvector analysis of random matrices with low expected rank". In: *arXiv preprint arXiv:1709.09565* (2017).

[68] Tatsuro Kawamoto and Yoshiyuki Kabashima. "Limitations in the spectral method for graph partitioning: Detectability threshold and localization of eigenvectors". In: *Physical Review E* 91.6 (2015), p. 062803.

[69] Tatsuro Kawamoto and Yoshiyuki Kabashima. "Detectability of the spectral method for sparse graph partitioning". In: *EPL (Europhysics Letters)* 112.4 (2015), p. 40007.

[70] Ludovic Stephan and Laurent Massoulié. "Robustness of spectral methods for community detection". In: *arXiv preprint arXiv:1811.05808* (2018).

[71] Zhenguo Li et al. "Noise robust spectral clustering". In: *2007 IEEE 11th International Conference on Computer Vision*. IEEE. 2007, pp. 1–8.

[72] Sivaraman Balakrishnan et al. "Noise thresholds for spectral clustering". In: *Advances in Neural Information Processing Systems*. 2011, pp. 954–962.

[73] Emmanuel Abbe et al. "Graph powering and spectral robustness". In: *arXiv preprint arXiv:1809.04818* (2018).

[74] Pan Zhang. "Robust spectral detection of global structures in the data by learning a regularization". In: *Advances in Neural Information Processing Systems*. 2016, pp. 541–549.

[75] Gergely Palla et al. "Uncovering the overlapping community structure of complex networks in nature and society". In: *nature* 435.7043 (2005), pp. 814–818.

[76] Yong-Yeol Ahn, James P Bagrow, and Sune Lehmann. "Link communities reveal multiscale complexity in networks". In: *nature* 466.7307 (2010), pp. 761–764.

[77] Austin R Benson, David F Gleich, and Jure Leskovec. "Higher-order organization of complex networks". In: *Science* 353.6295 (2016), pp. 163–166.

[78] Tiago P Peixoto. "Model selection and hypothesis testing for large-scale network models with overlapping groups". In: *Physical Review X* 5.1 (2015), p. 011033.

[79] Fei Wang et al. "Community discovery using nonnegative matrix factorization". In: *Data Mining and Knowledge Discovery* 22.3 (2011), pp. 493–521.

[80] Ioannis Psorakis et al. "Overlapping community detection using bayesian non-negative matrix factorization". In: *Physical Review E* 83.6 (2011), p. 066114.

[81] Jaewon Yang and Jure Leskovec. "Overlapping community detection at scale: a nonnegative matrix factorization approach". In: *Proceedings of the sixth ACM international conference on Web search and data mining*. 2013, pp. 587–596.

[82] Martin Rosvall et al. "Memory in network flows and its effects on spreading dynamics and community detection". In: *Nature communications* 5.1 (2014), pp. 1–13.

[83] Tiago P Peixoto. "Nonparametric Bayesian inference of the microcanonical stochastic block model". In: *Physical Review E* 95.1 (2017), p. 012317.

[84] Tiago P Peixoto. "The graph-tool python library. figshare (2014)". In: *DOI: https://doi.org/10.6084/m9. figshare* 1164194 (2014).

[85] Amr Ahmed et al. "Distributed large-scale natural graph factorization". In: *Proceedings of the 22nd international conference on World Wide Web*. 2013, pp. 37–48.

[86] Yoshiyuki Kabashima and Hisanao Takahashi. "First eigenvalue/eigenvector in sparse random symmetric matrices: influences of degree fluctuation". In: *Journal of Physics A: Mathematical and Theoretical* 45.32 (2012), p. 325001.

[87] Giulio Biroli and Rémi Monasson. "A single defect approximation for localized states on random lattices". In: *Journal of Physics A: Mathematical and General* 32.24 (1999), p. L255.

[88] Chihiro Noguchi and Tatsuro Kawamoto. "Fragility of spectral clustering for networks with an overlapping structure". In: *arXiv preprint arXiv:2003.02463* (2020).

Appendix A

Benchmark datasets

We performed numerical experiments on three different benchmark datasets as follows: the MovieLens 1M, 10M, and 20M datasets (https://movielens.org/). The characteristics of each dataset is represented in Table A.1.

Dataset	Rating set	#Users	#Items	#Ratings
MovieLens 1M	{1,2,3,4,5}	6,040	3,900	1,000,209
MovieLens 10M	{0.5,1,1.5,2,2.5,3,3.5,4,4.5,5}	10,681	71,567	10,000,054
MovieLens 20M	{0.5,1,1.5,2,2.5,3,3.5,4,4.5,5}	138,493	27,278	20,000,263

Table A.1: The details of the datasets used in this study. MovieLens is a dataset that consists of the ratings for movies from users who watched the movies, and the ratings of 1M dataset takes an integer value from 1 to 5 and those of 10M and 20M datasets take a value from 0.5 to 5 with step 0.5. When a user likes a movie very much, he or she rates the movie as 5. #Users and #Items correspond to the row and column sizes of the observed matrix, respectively, and #Ratings denotes the number of observations.

Appendix B

Derivation of the spectrum and the detectability limit of the canonical SBM

The goal of this appendix is to derive saddle-point expression of the average largest eigenvalue (4.16). Note that a similar calculation using the replica method can be found in Refs. [86, 68, 69]. We start with the average of nth moment of the partition function

$$[Z^n(M,\beta)]_M = \int \left(\prod_{a=1}^{n} d\boldsymbol{x}_a \delta(\boldsymbol{x}_a^\top \boldsymbol{x}_a - N)\right) \left[\exp\left(\frac{\beta}{2}\sum_a \boldsymbol{x}_a^\top M \boldsymbol{x}_a\right)\right]_M \tag{B.1}$$

$$= \int \left(\prod_{a=1}^{n} d\boldsymbol{x}_a \delta(\boldsymbol{x}_a^\top \boldsymbol{x}_a - N)\right) \left[\exp\left(-\frac{\beta}{2}\sum_a \left(\boldsymbol{\gamma}^\top \boldsymbol{x}_a\right)^2\right) \exp\left(\frac{\beta}{2}\sum_a \boldsymbol{x}_a^\top A \boldsymbol{x}_a\right)\right]_A, \tag{B.2}$$

where $\gamma_i \equiv d_i/\sqrt{2m}$. Introducing order parameters

$$\Omega_a = \frac{1}{\sqrt{N}}\sum_i \gamma_i x_{ia}, \quad (a \in \{1,\ldots,n\}), \tag{B.3}$$

we can recast exponential factor $e^{-\frac{\beta}{2}\sum_a (\boldsymbol{\gamma}^\top \boldsymbol{x}_a)^2}$ in (B.2) as

$$\exp\left(-\frac{\beta}{2}\sum_a \left(\boldsymbol{\gamma}^\top \boldsymbol{x}_a\right)^2\right) = \int \prod_a d\Omega_a \delta\left(\Omega_a - \frac{1}{\sqrt{N}}\sum_i \gamma_i x_{ia}\right) \exp\left(-\frac{\beta N}{2}\sum_a \Omega_a^2\right) \tag{B.4}$$

$$= \int \prod_a \frac{\beta N}{2\pi} d\Omega_a d\hat{\Omega}_a e^{-\frac{\beta N}{2}\sum_a (\Omega_a^2 - 2\Omega_a\hat{\Omega}_a)} e^{-\beta\sqrt{N}\sum_{ia}\hat{\Omega}_a\gamma_i x_{ia}} \tag{B.5}$$

$$= \int \prod_a \frac{\beta N}{2\pi} d\Omega_a d\hat{\Omega}_a e^{-\frac{\beta N}{2}\sum_a (\Omega_a^2 - 2\Omega_a\hat{\Omega}_a)} \prod_{ij} e^{-\frac{\beta}{\sqrt{c}}\sum_a \hat{\Omega}_a A_{ij} x_{ia}}. \tag{B.6}$$

We have set $\bar{c} \equiv 2m/N$. Moreover, $\hat{\Omega}_a$ is the auxiliary variable that is conjugate to Ω_a. To derive this expression, we transformed the delta function to

$$\delta\left(\sqrt{N}\Omega_a - \sum_i \gamma_i x_{ia}\right) = \int_{-i\infty}^{+i\infty} \frac{\beta\sqrt{N}}{2\pi} d\hat{\Omega}_a e^{\beta\sqrt{N}\hat{\Omega}_a(\sqrt{N}\Omega_a - \sum_i \gamma_i x_{ia})}. \tag{B.7}$$

Inserting Eq. (B.6) into the exponential factor in (B.2), we obtain

$$\begin{aligned}
&\left[\exp\left(\frac{\beta}{2}\sum_a \boldsymbol{x}_a^\top M \boldsymbol{x}_a\right)\right]_M \\
&= \int \prod_a \frac{\beta N}{2\pi} d\Omega_a d\hat{\Omega}_a e^{-\frac{\beta N}{2}(\Omega_a^2 - 2\Omega_a\hat{\Omega}_a)} \prod_{i<j} \sum_{A_{ij}\in\{0,1\}} \rho_{t_i t_j}^{A_{ij}} (1-\rho_{t_i t_j})^{1-A_{ij}} e^{\beta \sum_a A_{ij} x_{ia}\left(x_{ja} - \frac{2\hat{\Omega}_a}{\sqrt{\bar{c}}}\right)} \qquad \text{(B.8)} \\
&\approx \int \prod_a \frac{\beta N}{2\pi} d\Omega_a d\hat{\Omega}_a e^{-\frac{\beta N}{2}(\Omega_a^2 - 2\Omega_a\hat{\Omega}_a)} \prod_{i<j} \exp\left(\log(1-\rho_{t_i t_j}) + \rho_{t_i t_j} e^{\beta \sum_a x_{ia}(x_{ja} - \frac{2\hat{\Omega}_a}{\sqrt{\bar{c}}})}\right). \qquad \text{(B.9)}
\end{aligned}$$

Here, we took the configuration average over the canonical SBM (4.1) and approximated $\frac{\rho_{t_i t_j}}{1-\rho_{t_i t_j}} \approx \rho_{t_i t_j}$ by using the fact that $\rho_{t_i t_j} = O(N^{-1})$.

Let us now introduce the order-parameter functions

$$Q_k(\boldsymbol{u}) = \frac{1}{p_k N} \sum_{i\in V_k} \prod_a \delta(u_a - x_{ia}), \quad (k \in \{1, \dots, K\}) \tag{B.10}$$

where $\sum_{i\in V_k}$ is the sum over the node indices that belong to the kth block. Then, the last exponential factor in (B.9) can be approximated as

$$\exp\left(\sum_{i<j} \rho_{t_i t_j} e^{\beta \sum_a x_{ia}(x_{ja} - \frac{2\hat{\Omega}_a}{\sqrt{\bar{c}}})}\right) \approx \exp\left(\frac{N^2}{2} \int \prod_a du_a dv_a e^{\beta u_a\left(v_a - \frac{2\hat{\Omega}_a}{\sqrt{\bar{c}}}\right)} \sum_{kk'} Q_k(\boldsymbol{u}) W_{kk'} Q_{k'}(\boldsymbol{u})\right), \tag{B.11}$$

where we approximated that the contribution from the diagonal elements is negligible, and we defined $W_{kk'} \equiv p_k \rho_{kk'} p_{k'}$. Inserting Eq. (B.11) into (B.9), Eq. (B.1) is now expressed as

$$\begin{aligned}
[Z^n(M,\beta)]_M = &\int \prod_a d\boldsymbol{x}_a \int \prod_a \frac{\beta N}{2\pi} d\hat{\Omega}_a d\Omega_a \prod_a \delta(\boldsymbol{x}_a \boldsymbol{x}_a^\top - N) \\
&\times \exp\left(-\frac{\beta N}{2}\sum_a (\Omega_a^2 - 2\Omega_a\hat{\Omega}_a) + \sum_{i<j} \log(1-\rho_{t_i t_j})\right. \\
&\left. + \frac{N^2}{2} \int \prod_a du_a dv_a e^{\beta u_a\left(v_a - \frac{2\hat{\Omega}_a}{\sqrt{\bar{c}}}\right)} \sum_{kk'} Q_k(\boldsymbol{u}) W_{kk'} Q_{k'}(\boldsymbol{u})\right). \qquad \text{(B.12)}
\end{aligned}$$

Here, we use the expansion of the delta function

$$\delta\left(\boldsymbol{x}_a^\top \boldsymbol{x}_a - N\right) = \int_{-i\infty}^{+i\infty} \frac{\beta d\phi_a}{4\pi} e^{-\frac{\beta}{2}\phi_a\left(\sum_i x_{ia}^2 - N\right)} \tag{B.13}$$

and the identity

$$1 = \prod_k p_k N \int \frac{DQ_k}{2\pi} \delta \left(\sum_{i \in V_k} z_i \prod_{a=1}^{n} \delta(x_{ia} - \mu_a) - p_k N Q_k(\boldsymbol{\mu}) \right) \tag{B.14}$$

$$= \prod_k p_k N \int \frac{DQ_k D\hat{Q}_k}{2\pi} \exp \left(\sum_k \int d\boldsymbol{\mu} \hat{Q}_k(\boldsymbol{\mu}) \left(\sum_{i \in V_k} z_i \prod_{a=1}^{n} \delta(x_{ia} - \mu_a) - p_k N Q_k(\boldsymbol{\mu}) \right) \right). \tag{B.15}$$

Here, $\int DQ_k$ is the functional integral with respect to $Q_k(\boldsymbol{\mu})$, and $\hat{Q}_k(\boldsymbol{\mu})$ was introduced as the conjugate of $Q_k(\boldsymbol{\mu})$. To derive Eq. (B.15), we used the expansion of the delta function. By inserting the identity, we can focus on $Q_k(\boldsymbol{\mu})$ corresponding to the replacement in (B.10). Note that without the insertion of the identity, the replacement of (B.10) becomes invalid. From these, we can recast Eq. (B.12) as

$$\begin{aligned}
[Z^n(M,\beta)]_M = & \int \prod_a \frac{d\phi_a}{4\pi} \int \prod_a \frac{\beta N}{2\pi} d\hat{\Omega}_a d\Omega_a \int \prod_k \frac{p_k N}{2\pi} D\hat{Q}_k DQ_k \\
& \times \exp \left(-\frac{\beta N}{2} \sum_a (\Omega_a^2 - 2\Omega_a \hat{\Omega}_a - \phi_a) + \sum_{i<j} \log(1 - \rho_{t_i t_j}) \right. \\
& \left. - \sum_k p_k N L_k(Q_k, \hat{Q}_k) + K(\{Q_k\}) + \sum_k \sum_{i \in V_k} \log M_{i,k}(\hat{Q}_k, \{\phi_a\}) \right),
\end{aligned} \tag{B.16}$$

where

$$K(\{Q_k\}) = \frac{N^2}{2} \int \prod_a du_a dv_a e^{\beta u_a \left(v_a - \frac{2\hat{\Omega}_a}{\sqrt{\tilde{c}}} \right)} \sum_{kk'} Q_k(\boldsymbol{u}) W_{kk'} Q_{k'}(\boldsymbol{v}), \tag{B.17}$$

$$L_k(Q_k, \hat{Q}_k) = \int d\boldsymbol{u} Q_k(\boldsymbol{u}) \hat{Q}_k(\boldsymbol{u}), \tag{B.18}$$

$$M_{i,k}(\hat{Q}_k, \{\phi_a\}) = \int \prod_a d\boldsymbol{x}_a e^{\hat{Q}_k(\boldsymbol{x}_i) - \frac{\beta}{2} \sum_a \phi_a x_{ia}^2}. \tag{B.19}$$

Here, we assume the functional form of $Q_k(\boldsymbol{u})$ and $\hat{Q}_k(\boldsymbol{u})$ are restricted to Gaussian mixtures. This indicates that $Q_k(\boldsymbol{u})$ and $\hat{Q}_k(\boldsymbol{u})$ can be expressed as

$$Q_k(\boldsymbol{u}) = q_k^0 \int dA dH q_k(A,H) \left(\frac{\beta A}{2\pi} \right)^{\frac{n}{2}} \exp \left(-\frac{\beta A}{2} \sum_a \left(\mu_a - \frac{H}{A} \right)^2 \right), \tag{B.20}$$

$$\hat{Q}_k(\boldsymbol{u}) = \hat{q}_k^0 \int d\hat{A} d\hat{H} \hat{q}_k(\hat{A}, \hat{H}) \exp \left(\frac{\beta}{2} \sum_a \left(\hat{A} \mu_a^2 + 2\hat{H} \mu_a \right) \right), \tag{B.21}$$

where $q_k(A,H)$ is the weight of a Gaussian distribution with the mean and precision parameter equal to H/A and H, respectively. $\hat{q}_k(\hat{A}, \hat{H})$ is defined analogously. q_k^0 and $\hat{q}_k^0$ are the

normalization constants; it can be deduced that $q_k^0 = 1$ and $\hat{q}_k^0 = c_k$ from the saddle-point conditions when $n = 0$. Inserting Eq. (B.20) and (B.21) into (B.17)–(B.19), we have

$$K(\{Q_k\}) = \frac{N^2}{2} \sum_{kk'} W_{kk'} \int dA dH q_k(A, H) \int dA' dH' q_{k'}(A', H') \left(\frac{AA'}{AA' - 1} \right)^{-\frac{n}{2}} \times \exp \left[\sum_a \frac{\beta}{2} \left(\frac{A' \left(H - \frac{2\hat{\Omega}_a}{\sqrt{\bar{c}}} \right)^2 + 2 \left(H - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}} \right) H' + AH'^2}{AA' - 1} - \frac{\left(H - \frac{2\hat{\Omega}_a}{\sqrt{\bar{c}}} \right)^2}{A} - \frac{H'^2}{A} \right) \right], \tag{B.22}$$

$$L_k(Q_k, \hat{Q}_k) = \int dA dH d\hat{A} \hat{H} q_k(A, H) \hat{q}_k(\hat{A}, \hat{H}) \left(\frac{A}{A - \hat{A}} \right)^{\frac{n}{2}} \exp \left[\frac{n\beta}{2} \left(\frac{(H + \hat{H})^2}{A - \hat{A}} - \frac{H^2}{A} \right) \right], \tag{B.23}$$

$$M_{i,k}(\hat{Q}_k, \{\phi_a\}) = \left(\frac{2\pi}{\beta} \right)^{\frac{n}{2}} \sum_{d=0}^{\infty} \frac{c_k^d}{d!} \int \prod_{g=1}^{d} \left(d\hat{A}_g d\hat{H}_g \hat{q}_k(\hat{A}_g, \hat{H}_g) \right) \prod_a \left(\phi_a - \sum_g \hat{A}_g \right)^{-\frac{1}{2}} \times \exp \left(\frac{\beta}{2} \frac{\left(\sum_g \hat{H}_g \right)^2}{\phi_a - \sum_g \hat{A}_g} \right). \tag{B.24}$$

To derive Eq. (B.24), we expanded the exponential as $e^{\hat{Q}_k(\boldsymbol{x}_i)} = \sum_{d=0}^{\infty} \frac{1}{d!} \hat{Q}_k^d(\boldsymbol{x}_i)$.

Hereafter, let us assume no distinction among the variables with different replica indices, i.e., $\phi_a = \phi$, $\Omega_a = \Omega$, and $\hat{\Omega}_a = \hat{\Omega}$. This is referred to as the *replica symmetric assumption*. We insert Eq. (B.22)–(B.24) into (B.16) under this assumption. Then, we obtain the following saddle-point equation for the average largest eigenvalue from Eqs. (4.10), (4.11), and (4.14) as

$$[\lambda(M)]_M = 2 \lim_{\beta \to \infty} \frac{1}{\beta N} \lim_{n \to 0} \frac{\partial}{\partial n} \log [Z^n]_M \tag{B.25}$$

$$= \underset{\phi, \Omega, \hat{\Omega}, \{q_k, \hat{q}_k\}}{\text{extr}} \left[\phi + 2\Omega\hat{\Omega} - \Omega^2 + \frac{1}{2} \sum_{kk'} N W_{kk'} \int dA dH q_k(A, H) \int dA' dH' q_{k'}(A', H') \times \left(\frac{A' \left(H - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}} \right)^2 + 2 \left(H - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}} \right) H' + AH'^2}{AA' - 1} - \frac{(H - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}})^2}{A} - \frac{H'^2}{A'} \right) \right.$$
$$- \sum_k p_k c_k \int dA dH d\hat{A} d\hat{H} q_k(A, H) \hat{q}_k(\hat{A}, \hat{H}) \left(\frac{(H + \hat{H})^2}{A - \hat{A}} - \frac{H^2}{A} \right)$$
$$\left. + \sum_k p_k \sum_{d=0}^{\infty} \mathcal{P}_{c_k}(d) \int \prod_{g=1}^{d} (d\hat{A}_g d\hat{H}_g \hat{q}_k(\hat{A}_g, \hat{H}_g)) \frac{\left(\sum_g \hat{H}_g \right)^2}{\phi - \sum_g \hat{A}_g} \right]. \tag{B.26}$$

Here, $\mathcal{P}_{c_k}(d)$ is the probability mass function of degree d of each node in block k that has expectation c_k. From the saddle-point condition in Eq. (B.26), we obtain the functional equations with respect to $q_k(A,H)$ and $\hat{q}_k(\hat{A},\hat{H})$ as

$$q_k(A,H) = \sum_{d=0}^{\infty} \mathcal{P}_{c_k}(d) d \int \prod_{g=1}^{d-1} \left(d\hat{A}_g d\hat{H}_g \hat{q}_k(\hat{A}_g, \hat{H}_g) \right) \delta\left(H - \sum_{g=1}^{d-1} \hat{H}_g \right) \delta\left(A - \phi + \sum_{g=1}^{d-1} \hat{A}_g \right), \tag{B.27}$$

$$\hat{q}_k(\hat{A},\hat{H}) = \frac{1}{c_k} \sum_{k'} N \rho_{kk'} p_{k'} \int dA' dH' q_{k'}(A',H') \delta\left(\hat{A} - \frac{1}{A'} \right) \delta\left(\hat{H} - \frac{H' - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}}}{A'} \right). \tag{B.28}$$

To derive Eq. (B.27), we used the fact that the expectation of H^2/A becomes 0, which is derived by substituting $\hat{H} = \hat{A} = 0$. Moreover, the saddle-point condition with respect to ϕ yields

$$\sum_k p_k \int dA dH \mathcal{Q}_k(A,H) \left(\frac{H}{A} \right)^2 = 1, \tag{B.29}$$

where

$$\mathcal{Q}_k(A,H) = \sum_{d=0}^{\infty} \mathcal{P}_{c_k}(d) \int \prod_{g=1}^{d} \left(d\hat{A}_g d\hat{H}_g \hat{q}_k(\hat{A}_g, \hat{H}_g) \right) \delta\left(H - \sum_{g=1}^{d} \hat{H}_g \right) \delta\left(A - \phi + \sum_{g=1}^{d} \hat{A}_g \right). \tag{B.30}$$

Equation (B.29) corresponds to the normalization constraint in (4.9). Equations (B.27) and (B.28) constitute functional equations under constraint (B.29), and solving these equations yields the distribution of the largest eigenvector elements. Note that $q_k(A,H)$ was introduced as the weight in the Gaussian mixture, which approximates the empirical distribution of the largest eigenvector elements in (B.10). This indicates that $q_k(A,H)$ exhibits the probability density of the eigenvector-element distribution.

Unfortunately, solving the functional form of equations is still not analytically tractable. Thus, we introduce further approximations that $q_k(A) = \delta(A - a_k)$ and $\hat{q}_k(\hat{A}) = \delta(\hat{A} - \hat{a}_k)$, i.e., we ignore the fluctuation of the precision parameters. This is called the *effective medium*

approximation (EMA) [87, 86]. Performing the EMA for (B.26), we arrive at

$$
\begin{aligned}
[\lambda(M)]_M = \underset{\phi,\Omega,\hat{\Omega},m_{1k},m_{2k},\hat{m}_{1k},\hat{m}_{2k},a_k,\hat{a}_k}{\mathrm{extr}} \Bigg\{ & \phi + 2\hat{\Omega}\Omega - \Omega^2 \\
& + \frac{1}{2}N\sum_{k,k'} W_{kk'} \left(\frac{a_{k'}\left(m_{2k} - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}} + \frac{4\hat{\Omega}^2}{\bar{c}}\right) + 2m_{1k'}\left(m_{1k} - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}}\right) + a_k m_{2k'}}{a_k a_{k'} - 1} \right. \\
& \qquad \left. - \frac{m_{2k} - \frac{2\hat{\Omega}}{\sqrt{\bar{c}}}m_{1k} + \frac{4\hat{\Omega}}{\bar{c}}}{a_k} - \frac{m_{2k'}}{a_{k'}} \right) \\
& - \sum_k p_k c_k \left(\frac{m_{2k} + 2m_{1k}\hat{m}_{1k} + \hat{m}_{2k}}{a_k - \hat{a}_k} - \frac{m_{2k'}}{a_{k'}} \right) \\
& + \frac{1}{N}\sum_k \sum_{i\in V_k} \sum_{d=0}^{\infty} \frac{\mathcal{P}_{c_k}(d)}{\phi - d\hat{a}_k} \left(d\hat{m}_{2k} + d(d-1)\hat{m}_{1k}^2 \right) \Bigg\},
\end{aligned}
\tag{B.31}
$$

where $m_{\ell k}$ and $\hat{m}_{\ell k}$ stand for the ℓth moments of H and $\hat{H}$, respectively, i.e., $m_{\ell k} = \int dH H^\ell q_k(H)$ and $\hat{m}_{\ell k} = \int d\hat{H}\hat{H}^\ell \hat{q}_k(\hat{H})$.

The saddle-point conditions from (B.31) lead to the equations for the auxiliary variables $\phi, \Omega, \hat{\Omega}, m_{\ell k}, \hat{m}_{\ell k}, a_k$, and $\hat{a}_k$. Here, we focus on a model with the symmetry between the community blocks: $p_1 = p_3$ and $c_1 = c_3$. Due to this assumption, we can apply the same assumptions to the physical quantities $a_k, \hat{a}_k, m_{2k}, \hat{m}_{2k}$, that is, $a_1 = a_3$, $\hat{a}_1 = \hat{a}_3$, $m_{21} = m_{23}$, and $\hat{m}_{21} = \hat{m}_{23}$. This is because these quantities are the second-order statistics and do not depend on the signs.

Further, we assume $m_{12} = 0$. This assumption stems from the fact that the overlapping block does not contain nodes in the community blocks. Thus, the corresponding elements of the eigenvector come from a random structure of the graph. Moreover, we classify the solution into the cases of $m_{11} = 0$ and $m_{11} \neq 0$. For the solution with $m_{11} = 0$, we can assume $m_{13} = 0$ owing to the symmetry. On the other hand, for the solutions with $m_{11} \neq 0$, we can assume $m_{11} = -m_{13}$ due to the symmetry and the fact that the eigenvector elements of $\boldsymbol{x}$ tend to have the same signs in the same block. In summary, we have two types of solutions: $m_{11} = -m_{13} \neq 0$, $m_{12} = 0$ and $m_{11} = m_{12} = m_{13} = 0$. In fact, the former corresponds to the detectable condition and the latter corresponds to the undetectable condition. The leading eigenvalue is calculated for each of the two conditions, and the detectability limit is derived as the boundary between these two conditions. We further simplify the problem using the regular approximation with respect to the degree, namely the random variables following the Poisson distribution d in (4.16) are fixed as their means c_k.

First, under the detectable condition, we can derive the equations for $a_1, a_2, \hat{a}_1$, and $\hat{a}_2$

from the saddle-point conditions as

$$a_1 + (c_1 - 1)\hat{a}_1 = a_2 + (c_2 - 1)\hat{a}_2, \tag{B.32}$$

$$\frac{1}{a_1 - \hat{a}_1} = \frac{1+\epsilon}{1+\alpha+\epsilon}\frac{a_1}{a_1^2 - 1} + \frac{\alpha}{1+\alpha+\epsilon}\frac{a_2}{a_1 a_2 - 1}, \tag{B.33}$$

$$\frac{1}{a_2 - \hat{a}_2} = \frac{\sigma\alpha}{\sigma\alpha+2}\frac{a_2}{a_2^2 - 1} + \frac{2}{\sigma\alpha+2}\frac{a_1}{a_1 a_2 - 1}. \tag{B.34}$$

$$\frac{1}{a_1 - \hat{a}_1} = \frac{1-\epsilon}{1+\epsilon+\alpha}\frac{c_1 - 1}{a_1^2 - 1}. \tag{B.35}$$

We let the solutions of Eq. (B.32)–(B.35) as $a_1^{\rm det}$, $a_2^{\rm det}$, $\hat{a}_1^{\rm det}$, and $\hat{a}_2^{\rm det}$. Then, we obtain the average leading eigenvalue as

$$[\lambda(M)]_M = \phi = a_k^{\rm det} + (c_k - 1)\hat{a}_k^{\rm det} \quad (k = 1, 2) \tag{B.36}$$

and the condition of the detectability limit as

$$D(a_1^{\rm det}, a_2^{\rm det}, \hat{a}_1^{\rm det}, \hat{a}_2^{\rm det}) = 0, \tag{B.37}$$

where

$$D(a_1, a_2, \hat{a}_1, \hat{a}_2) = M_{11}M_{22} - M_{12}M_{21}, \tag{B.38}$$

$$M_{11} = (1+\epsilon)\frac{a_1^2 + 1}{(a_1^2 - 1)^2} + \alpha\frac{a_2^2}{(a_1 a_2 - 1)^2} - (1+\alpha+\epsilon)\frac{1}{(a_1 - \hat{a}_1)^2}\frac{c_1}{c_1 - 1}, \tag{B.39}$$

$$M_{12} = \frac{\alpha}{(a_1 a_2 - 1)^2}, \tag{B.40}$$

$$M_{21} = \frac{2}{(a_1 a_2 - 1)^2}, \tag{B.41}$$

$$M_{22} = \frac{2a_1^2}{(a_1 a_2 - 1)^2} + \sigma\alpha\frac{a_2^2 + 1}{(a_2^2 - 1)^2} - (\sigma\alpha + 2)\frac{1}{(a_2 - \hat{a}_2)^2}\frac{c_2}{c_2 - 1}. \tag{B.42}$$

The detectability limit (B.37) is derived by condition $\hat{m}_{11}^2 = 0$, because $D(a_1, a_2, \hat{a}_1, \hat{a}_2)$ is proportional to $\hat{m}_{11}^2$.

Second, under the undetectable condition, we can derive the equations for $a_1, a_2, \hat{a}_1$, and $\hat{a}_2$ from the saddle-point conditions as

$$a_1 + (c_1 - 1)\hat{a}_1 = a_2 + (c_2 - 1)\hat{a}_2, \tag{B.43}$$

$$\frac{1}{a_1 - \hat{a}_1} = \frac{1+\epsilon}{1+\alpha+\epsilon}\frac{a_1}{a_1^2 - 1} + \frac{\alpha}{1+\alpha+\epsilon}\frac{a_2}{a_1 a_2 - 1}, \tag{B.44}$$

$$\frac{1}{a_2 - \hat{a}_2} = \frac{\sigma\alpha}{\sigma\alpha+2}\frac{a_2}{a_2^2 - 1} + \frac{2}{\sigma\alpha+2}\frac{a_1}{a_1 a_2 - 1}, \tag{B.45}$$

$$D(a_1, a_2, \hat{a}_1, \hat{a}_2) = 0. \tag{B.46}$$

These equations are analogous to those for the detectable conditions (B.32)–(B.35). A crucial difference is that we have condition $\hat{m}_{11}^2 = 0$ instead of Eq. (B.35). We let the solutions of these equations be $a_1^{\rm und}$, $a_2^{\rm und}$, $\hat{a}_1^{\rm und}$, and $\hat{a}_2^{\rm und}$. Using this solution, we obtain the average leading eigenvalue in the undetectable conditions as follows.

$$[\lambda(M)]_M = \phi = a_k^{\rm und} + (c_k - 1)\hat{a}_k^{\rm und}. \quad (k = 1, 2) \tag{B.47}$$

Appendix C

Microcanonical overlapping SBM

In this appendix, we discuss the microcanonical SBM. In Sec. C.1, we introduce the definition of the microcanonical overlapping SBM. In Sec. C.2, we provide the replica analysis to derive its spectrum and the detectability limit. In Sec. C.3, we derive the saddle-point conditions for normalization constant $\mathcal{N}_G$, from which we can derive crucial relations used in Sec. C.2. Finally, in Sec. C.4, we discuss the distinction between the canonical and microcanonical SBMs and discuss the reason of their use in our numerical experiments.

C.1 Model definition

Microcanonical SBM is an SBM that is formulated on the basis of different constraints from its canonical model. Although the canonical SBM specifies the expected number of edges within the blocks, the microcanonical SBM specifies the number of edges within the blocks as well as the degree sequence as hard constraints. The microcanonical SBM generates a graph uniformly and randomly from all realizable graphs under these constraints. We denote the sequence of node degrees as $\boldsymbol{d} = [d_i]$. We let e_{kl} be the number of edges between blocks k and l; we denote the corresponding matrix as $\boldsymbol{e} = [e_{kl}]$. Moreover, $\boldsymbol{t} = [t_i]$ $t_i \in \{1, \cdots, K\}$ $(i \in V)$ are the planted block labels of the nodes. An instance of the microcanonical SBM is generated according to the following probability distribution.

$$P(A|\boldsymbol{d}, \boldsymbol{e}, \boldsymbol{t}) = \frac{1}{\Omega(\boldsymbol{d}, \boldsymbol{e}, \boldsymbol{t})}, \tag{C.1}$$

where $\Omega(\boldsymbol{d}, \boldsymbol{e}, \boldsymbol{t})$ is the number of all realizable graphs under given $\boldsymbol{d}$, $\boldsymbol{e}$, and $\boldsymbol{t}$.

We consider a microcanonical SBM with an overlapping structure with the following parametrization.

$$\boldsymbol{p} = (p_1, p_2, p_3) = (p_1, \alpha p_1, p_1)\,, \tag{C.2}$$

$$\boldsymbol{e} = \begin{pmatrix} 1 & \alpha & \epsilon \\ \alpha & \sigma\alpha^2 & \alpha \\ \epsilon & \alpha & 1 \end{pmatrix} e_{11}, \tag{C.3}$$

$$d_i = c_{t_i}. \tag{C.4}$$

Although we can provide an arbitrary degree sequence, for simplicity, we assume the nodes belonging to the same block k have equal degree c_k. As in the canonical SBM, the model parameters must satisfy constraint (4.6).

C.2 Derivation of the spectrum and the detectability limit of the microcanonical SBM

Here, we conduct an analysis analogous to Appendix B for the microcanonical SBM. As a result of the present analysis, we obtain the same average largest eigenvalues as those of the canonical case in (B.36) and (B.47). However, a different technique is required to impose the microcanonical constraints. The calculations in this appendix are extensions of those in Refs. [68, 69]. We start with the nth moment of the partition function (4.15)

$$[Z^n(M,\beta)]_M = \int \left(\prod_{a=1}^{n} d\boldsymbol{x}_a \delta(\boldsymbol{x}_a^\top \boldsymbol{x}_a - N)\right) \left[\exp\left(\frac{\beta}{2}\sum_a \boldsymbol{x}_a^\top M \boldsymbol{x}_a\right)\right]_M . \tag{C.5}$$

As defined in Appendix C.1, we assume the three blocks model. Then, the exponential factor in (C.5) can be recast as

$$\begin{aligned}\boldsymbol{x}_a^\top M \boldsymbol{x}_a = &\sum_{ij\in V_1} u_{ij}x_{ia}x_{ja} + \sum_{ij\in V_2} y_{ij}x_{ia}x_{ja} + \sum_{ij\in V_3} u_{ij}x_{ia}x_{ja} \\ &+ 2\sum_{i\in V_1}\sum_{j\in V_2} v_{ij}x_{ia}x_{ja} + 2\sum_{i\in V_2}\sum_{j\in V_3} v_{ij}x_{ia}x_{ja} + 2\sum_{i\in V_1}\sum_{j\in V_3} w_{ij}x_{ia}x_{ja} - (\boldsymbol{\gamma}^\top \boldsymbol{x}_a)^2, \end{aligned} \tag{C.6}$$

where u_{ij}, y_{ij}, v_{ij}, and w_{ij} are the adjacency matrix elements. These parameters were introduced to distinguish blocks that obey different statistics. Again, the summation $\sum_{i\in V_k}$ is taken over indices of the nodes that belong to block k.

To calculate the ensemble average over the microcanonical SBM, we take the sum over all possible graph configurations as imposing the microcanonical constraints by delta functions.

Thus, the configuration average of the exponential factor in (C.5) is

$$
\begin{aligned}
&\left[\exp\left(\frac{\beta}{2}\sum_a \boldsymbol{x}_a^\top M \boldsymbol{x}_a\right)\right]_M \\
&= \frac{1}{\mathcal{N}_G}\sum_{\{u_{ij}\},\{w_{ij}\},\{v_{ij}\},\{w_{ij}\}}\prod_{i\in V_1}\delta\left(\sum_{l\in V_1}u_{il}+\sum_{m\in V_2}v_{im}+\sum_{n\in V_3}w_{in}-c_1\right) \\
&\times\prod_{j\in V_2}\delta\left(\sum_{l\in V_1}u_{jl}+\sum_{m\in V_2}v_{jm}+\sum_{n\in V_3}w_{jn}-c_2\right)\prod_{k\in V_3}\delta\left(\sum_{l\in V_1}u_{kl}+\sum_{m\in V_2}v_{km}+\sum_{n\in V_3}w_{kn}-c_3\right) \\
&\times\delta\left(\sigma p_2\sum_{i\in V_1}\sum_{j\in V_2}v_{ij}-p_1\sum_{i,j\in V_2}y_{ij}\right)\delta\left(\sigma p_2\sum_{i\in V_2}\sum_{j\in V_3}v_{ij}-p_3\sum_{i,j\in V_2}y_{ij}\right) \\
&\times\delta\left(p_2\sum_{i,j\in V_1}u_{ij}-p_1\sum_{i\in V_1}\sum_{j\in V_2}v_{ij}\right)\delta\left(p_2\sum_{i,j\in V_3}u_{ij}-p_3\sum_{i\in V_2}\sum_{j\in V_3}v_{ij}\right) \\
&\times\delta\left(\epsilon\sum_{i,j\in V_1}u_{ij}-\sum_{i\in V_1}\sum_{j\in V_3}w_{ij}\right)\delta\left(\epsilon\sum_{i,j\in V_3}u_{ij}-\sum_{i\in V_1}\sum_{j\in V_3}w_{ij}\right)\exp\left(\frac{\beta}{2}\sum_a \boldsymbol{x}_a^\top M\boldsymbol{x}_a\right). \qquad \text{(C.7)}
\end{aligned}
$$

Here, $\mathcal{N}_G$ is the number of all realizable graphs that satisfy the constraints. The first three delta functions in (C.7) represent Kronecker's deltas that impose the degree constraints, while the remaining ones represent Dirac's deltas that impose the constraints with respect to the number of edges between blocks, as specified by matrix $\boldsymbol{e}$.

We use the integral expression of the delta functions as follows.

$$\delta(x) = \oint\frac{dz}{2\pi}z^{x-1}, \qquad \text{(C.8)}$$

$$\delta(x) = \int_{-i\infty}^{i\infty}\frac{d\eta}{2\pi}e^{-\eta x}. \qquad \text{(C.9)}$$

Here, Eqs. (C.8) and (C.9) correspond to the Kronecker's and Dirac's deltas. Then, Eq. (C.7) can be recast as follows.

$$
\begin{aligned}
&\frac{1}{\mathcal{N}_G}\oint\prod_{k=1,2,3}\prod_{i\in V_k}\frac{dz_i}{2\pi}z_i^{-(1+c_k)}\int\frac{d\zeta}{2\pi}\int\frac{d\xi}{2\pi}\int\frac{d\tau}{2\pi}\int\frac{d\kappa}{2\pi}\int\frac{d\eta}{2\pi}\int\frac{d\theta}{2\pi}e^{-\frac{\beta}{2}\sum_a(\boldsymbol{\gamma}^\top\boldsymbol{x}_a)^2} \\
&\times\prod_{\substack{i<j\\ i,j\in V_1}}\sum_{u_{ij}\in\{0,1\}}\left(z_iz_je^{\beta\sum_a x_{ia}x_{ja}-2\tau p_2-2\eta\epsilon}\right)^{u_{ij}}\prod_{\substack{i<j\\ i,j\in V_2}}\sum_{y_{ij}\in\{0,1\}}\left(z_iz_je^{\beta\sum_a x_{ia}x_{ja}+2\xi p_3+2\zeta p_1}\right)^{y_{ij}} \\
&\times\prod_{\substack{i<j\\ i,j\in V_3}}\sum_{u_{ij}\in\{0,1\}}\left(z_iz_je^{\beta\sum_a x_{ia}x_{ja}-2\kappa p_2-2\theta\epsilon}\right)^{u_{ij}}\prod_{i\in V_1}\prod_{j\in V_2}\sum_{v_{ij}\in\{0,1\}}\left(z_iz_je^{\beta\sum_a x_{ia}x_{ja}-\sigma\zeta p_2+\tau p_1}\right)^{v_{ij}} \\
&\times\prod_{i\in V_2}\prod_{j\in V_3}\sum_{v_{ij}\in\{0,1\}}\left(z_iz_je^{\beta\sum_a x_{ia}x_{ja}-\sigma\xi p_2+\kappa p_3}\right)^{v_{ij}}\prod_{i\in V_1}\prod_{j\in V_3}\sum_{w_{ij}\in\{0,1\}}\left(z_iz_je^{\beta\sum_a x_{ia}x_{ja}+\eta+\theta}\right)^{w_{ij}}, \qquad \text{(C.10)}
\end{aligned}
$$

where parameters $\zeta, \xi, \tau, \kappa, \eta$, and θ are the auxiliary variables provided by the integral representation of the delta function. Because variables u_{ij}, y_{ij}, v_{ij}, and w_{ij} only take binary values, their summations in (C.10) can be calculated straightforwardly. For example,

$$\prod_{\substack{i<j \\ i,j\in V_1}} \sum_{u_{ij}\in\{0,1\}} \left(z_i z_j e^{\beta \sum_a x_{ia}x_{ja}}\right)^{u_{ij}} = \prod_{\substack{i<j \\ i,j\in V_2}} (1 + z_i z_j e^{\beta \sum_a x_{ia}x_{ja}}) \approx \prod_{\substack{i<j \\ i,j\in V_3}} \exp(z_i z_j e^{\beta \sum_a x_{ia}x_{ja}}). \tag{C.11}$$

To derive the last equation in (C.11), we assume that $|z_i|$ and $|z_j|$ are sufficiently small.

Here, we introduce the order-parameter functions

$$Q_k(\boldsymbol{\mu}) = \frac{1}{p_k N} \sum_{i\in V_k} z_i \prod_{a=1}^{n} \delta(x_{ia} - \mu_a), \quad (k = 1, 2, 3) \tag{C.12}$$

which is similar but not completely equivalent to (B.10). Using the order-parameter functions (C.12), when $N \gg 1$, Eq. (C.11) can be approximated as

$$\prod_{\substack{i<j \\ i,j\in V_1}} \exp(z_i z_j e^{\beta \sum_a x_{ia}x_{ja}}) \approx \exp\left(\frac{(p_1 N)^2}{2} \int \prod_{a=1}^{n} d\mu_a d\nu_a Q_1(\boldsymbol{\mu}) Q_1(\boldsymbol{\nu}) e^{\beta \sum_a \mu_a \nu_a}\right), \tag{C.13}$$

where we approximated that the contribution from the diagonal elements is negligible. Using the similar calculations, (C.5) is now written as

$$[Z^n(M, \beta)]_M = e^{N\mathcal{T}_n(Q) + N\mathcal{S}_n}, \tag{C.14}$$

where

$$\begin{aligned}
N\mathcal{T}_n(Q) = {} & \frac{(p_1 N)^2}{2} \int \prod_{a=1}^{n} d\mu_a d\nu_a Q_1(\boldsymbol{\mu}) Q_1(\boldsymbol{\nu}) e^{\beta \sum_a \mu_a \nu_a - 2\tau p_2 - 2\eta\epsilon} \\
& + \frac{(p_2 N)^2}{2} \int \prod_{a=1}^{n} d\mu_a d\nu_a Q_2(\boldsymbol{\mu}) Q_2(\boldsymbol{\nu}) e^{\beta \sum_a \mu_a \nu_a + 2\xi p_3 + 2\zeta p_1} \\
& + \frac{(p_3 N)^2}{2} \int \prod_{a=1}^{n} d\mu_a d\nu_a Q_3(\boldsymbol{\mu}) Q_3(\boldsymbol{\nu}) e^{\beta \sum_a \mu_a \nu_a - 2\kappa p_2 - 2\theta\epsilon} \\
& + p_1 p_2 N^2 \int \prod_{a=1}^{n} d\mu_a d\nu_a Q_1(\boldsymbol{\mu}) Q_2(\boldsymbol{\nu}) e^{\beta \sum_a \mu_a \nu_a - \sigma\zeta p_2 + 2\tau p_1} \\
& + p_2 p_3 N^2 \int \prod_{a=1}^{n} d\mu_a d\nu_a Q_2(\boldsymbol{\mu}) Q_3(\boldsymbol{\nu}) e^{\beta \sum_a \mu_a \nu_a - \sigma\xi p_2 + 2\kappa p_3} \\
& + p_1 p_3 N^2 \int \prod_{a=1}^{n} d\mu_a d\nu_a Q_1(\boldsymbol{\mu}) Q_3(\boldsymbol{\nu}) e^{\beta \sum_a \mu_a \nu_a + \eta + \theta}
\end{aligned} \tag{C.15}$$

and

$$e^{N\mathcal{S}_n} = \int \prod_{i=1}^{N}\prod_{a=1}^{n} dx_{ia} \prod_{a=1}^{n} \delta\left(\sum_{i=1}^{N} x_{ia}^2 - N\right) \int \sqrt{N} \prod_{a=1}^{n} d\Omega_a \delta\left(\sqrt{N}\Omega_a - \sum_i \gamma_i x_{ia}\right) e^{-\frac{\beta}{2}\Omega_a^2} \times \frac{1}{\mathcal{N}_G} \int \frac{d\zeta}{2\pi} \int \frac{d\xi}{2\pi} \int \frac{d\tau}{2\pi} \int \frac{d\kappa}{2\pi} \int \frac{d\eta}{2\pi} \int \frac{d\theta}{2\pi}. \tag{C.16}$$

Here, Ω_a is the order parameter defined in (B.3). As in the case of the canonical SBM in (B.14), for Eq. (C.16), we insert the identity

$$1 = \prod_{k=1,2,3} p_k N \int \frac{DQ_k}{2\pi} \delta\left(\sum_{i\in V_k} z_i \prod_{a=1}^{n} \delta(x_{ia} - \mu_a) - p_k N Q_k(\boldsymbol{\mu})\right) \tag{C.17}$$

$$= \prod_{k=1,2,3} p_k N \int \frac{DQ_k D\hat{Q}_k}{2\pi} \exp\left(\sum_{k=1,2,3} \int d\boldsymbol{\mu} \hat{Q}_k(\boldsymbol{\mu}) \left(\sum_{i\in V_k} z_i \prod_{a=1}^{n} \delta(x_{ia} - \mu_a) - p_k N Q_k(\boldsymbol{\mu})\right)\right). \tag{C.18}$$

In (C.17), we perform the functional integration over the space of function $Q_k(\boldsymbol{\mu})$. It is required to insert identity (C.17), because it indicates that we performed the replacement of a function in (C.12) by $Q_k(\boldsymbol{\mu})$. Furthermore, using the integral representation of the delta functions (B.7) and (B.13), we obtain

$$e^{N\mathcal{S}_n} = \int \prod_{k=1,2,3} p_k N \frac{DQ_k D\hat{Q}_k}{2\pi} \int \prod_a \frac{\beta d\phi_a}{4\pi} \int \prod_a \frac{\beta N d\Omega_a \hat{\Omega}_a}{2\pi} \int \frac{d\zeta}{2\pi} \int \frac{d\xi}{2\pi} \int \frac{d\tau}{2\pi} \int \frac{d\kappa}{2\pi} \int \frac{d\eta}{2\pi} \int \frac{d\theta}{2\pi}$$
$$\times \exp\left(-\log N_{\mathcal{G}} - N\sum_k K_k(Q_k, \hat{Q}_k) + \frac{\beta N}{2} \sum_a (2\Omega_a \hat{\Omega}_a - \Omega_a^2 + \phi_a) - \sum_k \log c_k! \right.$$
$$\left. + \sum_{k=1,2,3} \log L_k\left(\hat{Q}_k, \{\hat{\Omega}_a\}, \{\phi_a\}\right)\right), \tag{C.19}$$

where

$$K_k(Q_k, \hat{Q}_k) = p_k \int d\boldsymbol{\mu} Q_k(\boldsymbol{\mu}) \hat{Q}_k(\boldsymbol{\mu}), \tag{C.20}$$

$$L_k\left(\hat{Q}_k, \{\hat{\Omega}_a\}, \{\phi_a\}\right) = \int \prod_{i\in V_k} \prod_a dx_{ia} \prod_{i\in V_k} \left(\hat{Q}_k^{c_k}(\boldsymbol{x}_i) \exp\left(-\beta \sum_a \left(\sqrt{N}\hat{\Omega}_a \gamma_i x_{ia} + \frac{1}{2}\phi_a x_{ia}^2\right)\right)\right). \tag{C.21}$$

Here, we used the relation

$$\oint \frac{dz_i}{2\pi} z_i^{-(1+c_k)} e^{z_i \hat{Q}_k(\boldsymbol{x}_i)} = \oint \frac{dz_i}{2\pi} z_i^{-(1+c_k)} \sum_{m=0}^{\infty} \frac{1}{m!} \left(z_i \hat{Q}_k(\boldsymbol{x}_i)\right)^m \tag{C.22}$$

$$= \sum_{m=0}^{\infty} \frac{1}{m!} \hat{Q}_k(\boldsymbol{x}_i) \oint \frac{dz_i}{2\pi} z_i^{m-(1+c_k)} \tag{C.23}$$

$$= \frac{1}{c_k!} \hat{Q}_k^{c_k}(\boldsymbol{x}_i). \tag{C.24}$$

Now, the variable depending on the node index i only appears as $\boldsymbol{x}_i$. Hence, after the integral with respect to $\boldsymbol{x}_i$ is carried out in $L_k\left(\hat{Q}_k, \{\hat{\Omega}_a\}, \{\phi_a\}\right)$, Eq. (C.19) can be expressed only with integrals over the auxiliary variables ϕ_a, Ω_a, $\hat{\Omega}_a$, ζ, ξ, τ, κ, η, θ and functional integrals over $Q_k(\boldsymbol{\mu})$ and $\hat{Q}_k(\boldsymbol{\mu})$.

For further calculations, as in the case of the canonical SBM (Eqs. (B.20) and (B.21)), we assume the functional form of Q_k and $\hat{Q}_k$ are restricted to the Gaussian mixtures as follows.

$$Q_k(\boldsymbol{\mu}) = T_k \int dA dH q_k(A,H) \left(\frac{\beta A}{2\pi}\right)^{\frac{n}{2}} \exp\left(-\frac{\beta A}{2}\sum_a \left(\mu_a - \frac{H}{A}\right)^2\right), \tag{C.25}$$

$$\hat{Q}_k(\boldsymbol{\mu}) = \hat{T}_k \int d\hat{A} d\hat{H} \hat{q}_k(\hat{A},\hat{H}) \exp\left(\frac{\beta}{2}\sum_a \left(\hat{A}\mu_a^2 + 2\hat{H}\mu_a\right)\right), \tag{C.26}$$

where T_k and $\hat{T}_k$ represent the normalization constants. With these functional forms, we can calculate the integrals over $\boldsymbol{\mu}$ in (C.20) and $\boldsymbol{x}$ in (C.21). Then, we obtain the following expressions.

$$\begin{aligned} K_k(q_k,\hat{q}_k) = c_k p_k \int dA dH \int d\hat{A} d\hat{H} q_k(A,H)\hat{q}_k(\hat{A},\hat{H}) \\ \times \left(\frac{A}{A-\hat{A}}\right)^{\frac{n}{2}} \exp\left(\frac{n\beta}{2}\left(\frac{(H+\hat{H})^2}{A-\hat{A}} - \frac{H^2}{A}\right)\right), \end{aligned} \tag{C.27}$$

$$\begin{aligned} L_k\left(\hat{q}_k, \{\hat{\Omega}_a\}, \{\phi_a\}\right) = \hat{T}_k^{c_k}\left(\frac{2\pi}{\beta}\right)^{\frac{n}{2}} \int \prod_{g=1}^{c_k} (d\hat{A}_g d\hat{H}_g \hat{q}_k(\hat{A}_g,\hat{H}_g)) \prod_{a=1}^{n}\left(\phi_a - \sum_{g=1}^{c_k}\hat{A}_g\right)^{-\frac{1}{2}} \\ \times \exp\left(\frac{\beta}{2}\sum_{i\in V_k} \frac{(\sqrt{N}\hat{\Omega}_a\gamma_i - \sum_{g=1}^{c_k}\hat{H}_g)^2}{\phi_a - \sum_{g=1}^{c_k}\hat{A}_g}\right). \end{aligned} \tag{C.28}$$

In Appendix C.3, we solve for normalization constants T_k and $\hat{T}_k$. By using (C.54), we can replace $T_k\hat{T}_k$ with c_k. This is how we eliminated the normalization constants in Eq. (C.27). By inserting (C.25) and (C.26) in (C.15), we can calculate the integrals over $\boldsymbol{\mu}$ and obtain

$$\begin{aligned} \mathcal{T}_n = N \int dA dH \int dA' dH' \left(\frac{AA'}{A-A'}\right)^{\frac{n}{2}} \exp\left(\frac{n\beta}{2}\Xi(A,A',H,H')\right) \\ \times \left(\frac{c_1}{2}\frac{p_1^2}{p_1+p_2+\epsilon p_1} q_1(A,H)q_1(A',H') + \frac{c_2}{2}\frac{\sigma p_2^2}{p_1+\sigma p_2+p_3} q_2(A,H)q_2(A',H')\right. \\ + \frac{c_3}{2}\frac{p_3^2}{p_3+p_2+\epsilon p_3} q_3(A,H)q_3(A',H') + c_2\frac{p_1 p_2}{p_1+\sigma p_2+p_3} q_1(A,H)q_2(A',H') \\ \left. + c_2\frac{p_2 p_3}{p_1+\sigma p_2+\sigma p_3} q_2(A,H)q_3(A',H') + c_1\frac{\epsilon p_1^2}{p_1+p_2+\epsilon p_1} q_1(A,H)q_3(A',H')\right), \end{aligned} \tag{C.29}$$

where

$$\Xi(A, A', H, H') = \frac{A'H^2 + AH'^2 + 2HH'}{AA' - 1} - \frac{H^2}{A} - \frac{H'^2}{A'}. \tag{C.30}$$

Here, we used the relations between T_1, T_2, and T_3 (C.55)–(C.61). From the calculations so far, we have performed all the integrals over $\boldsymbol{z}$, $\boldsymbol{x}$, and $\boldsymbol{\mu}$. The functional integrals over $Q_k(\boldsymbol{\mu})$ and $\hat{Q}_k(\boldsymbol{\mu})$ in (C.19) have been replaced by the integral over the functions $q_k(A, H)$ and $\hat{q}_k(\hat{A}, \hat{H})$. In summary, the nth moment of the partition function (C.14) is now represented by the integrals with respect to auxiliary variables ϕ_a, Ω_a, and $\hat{\Omega}_a$ and the functional integrals over $q_k(A, H)$ and $\hat{q}_k(\hat{A}, \hat{H})$. Note that the other variables ζ, ξ, τ, κ, η, and θ can be erased when inserting the relations between the normalization constants (C.55)–(C.61).

Again, as we assumed in the canonical SBM, we impose the replica symmetric assumptions for the parameters ϕ_a, Ω_a, and $\hat{\Omega}_a$, i.e., $\phi_a = \phi$, $\Omega_a = \Omega$, and $\hat{\Omega}_a = \hat{\Omega}$ in Eq. (C.27)–(C.29). Inserting Eq. (C.27)–(C.29) under the assumptions into (C.14) and taking the limit $N \to \infty$, the average largest eigenvalue can be expressed as follows.

$$\begin{aligned}
[\lambda(M)]_M \quad &= 2 \lim_{\beta\to\infty} \frac{1}{\beta N} \lim_{n\to 0} \frac{\partial}{\partial n} \log[Z^n]_M \qquad \text{(C.31)} \\
&= \underset{q_k, \hat{q}_k, \phi, \Omega, \hat{\Omega}}{\text{extr}} \left\{ \int dA dH \int dA' dH' \Xi(A, A', H, H') \right. \\
&\quad \times \left(\frac{c_1}{2} \frac{p_1^2}{p_1 + p_2 + \epsilon p_1} q_1(A, H) q_1(A', H') + \frac{c_2}{2} \frac{\sigma p_2^2}{p_1 + \sigma p_2 + p_3} q_2(A, H) q_2(A', H') \right. \\
&\quad + \frac{c_3}{2} \frac{p_3^2}{p_3 + p_2 + \epsilon p_3} q_3(A, H) q_3(A', H') + c_2 \frac{p_1 p_2}{p_1 + \sigma p_2 + p_3} q_1(A, H) q_2(A', H') \\
&\quad \left. + c_2 \frac{p_2 p_3}{p_1 + \sigma p_2 + p_3} q_2(A, H) q_3(A', H') + c_1 \frac{\epsilon p_1^2}{p_1 + p_2 + \epsilon p_1} q_1(A, H) q_3(A', H') \right) \\
&\quad - \sum_{k=1,2,3} c_k p_k \int dA dH \int d\hat{A} d\hat{H} q_k(A, H) \hat{q}_k(\hat{A}, \hat{H}) \left(\frac{(H + \hat{H})^2}{A - \hat{A}} - \frac{H^2}{A} \right) \\
&\quad + 2\Omega\hat{\Omega} - \Omega^2 + \phi \\
&\quad \left. + \frac{1}{N} \sum_{k=1,2,3} \int \prod_{g=1}^{c_k} \left(d\hat{A}_g d\hat{H}_g \hat{q}_k(\hat{A}_g, \hat{H}_g) \right) \sum_{i \in V_k} \frac{\left(\sqrt{N}\hat{\Omega}\gamma_i - \sum_{g=1}^{c_k} \hat{H}_g \right)^2}{\phi - \sum_{g=1}^{c_k} \hat{A}_g} \right\}. \qquad \text{(C.32)}
\end{aligned}$$

From Eq. (C.32), we obtain the saddle-point conditions as

$$\hat{q}_1(\hat{A},\hat{H}) = \int dA'dH' \frac{p_1 q_1(A',H') + p_2 q_2(A',H') + \epsilon p_1 q_3(A',H')}{p_1+p_2+\epsilon p_1} \delta\left(\hat{A} - \frac{1}{A'}\right)\delta\left(\hat{H} - \frac{H'}{A'}\right), \tag{C.33}$$

$$\hat{q}_2(\hat{A},\hat{H}) = \int dA'dH' \frac{p_1 q_1(A',H') + \sigma p_2 q_2(A',H') + p_3 q_3(A',H')}{p_1+\sigma p_2+p_3} \delta\left(\hat{A} - \frac{1}{A'}\right)\delta\left(\hat{H} - \frac{H'}{A'}\right), \tag{C.34}$$

$$\hat{q}_3(\hat{A},\hat{H}) = \int dA'dH' \frac{p_3 q_1(A',H') + p_2 q_2(A',H') + \epsilon p_3 q_3(A',H')}{p_3+p_2+\epsilon p_3} \delta\left(\hat{A} - \frac{1}{A'}\right)\delta\left(\hat{H} - \frac{H'}{A'}\right), \tag{C.35}$$

and

$$q_k(A,H) = \frac{1}{p_k N} \int \prod_{g=1}^{c_k-1} \left(d\hat{A}_g d\hat{H}_g \hat{q}_k(\hat{A}_g,\hat{H}_g)\right) \delta\left(H - \sum_{g=1}^{c_k-1}\hat{H}_g + \sqrt{N}\hat{\Omega}\gamma_i\right)\delta\left(A - \phi + \sum_{g=1}^{c_k-1}\hat{A}_g\right). \tag{C.36}$$

Moreover, the saddle-point conditions with respect to ϕ yield

$$\sum_k p_k \int dA dH \mathcal{Q}_k(A,H)\left(\frac{H}{A}\right)^2 = 1, \tag{C.37}$$

where

$$\mathcal{Q}_k(A,H) = \frac{1}{p_k N}\sum_{i\in V_k} \int \prod_{g=1}^{c_k} \left(d\hat{A}_g d\hat{H}_g \hat{q}_k(\hat{A}_g,\hat{H}_g)\right) \delta\left(H - \sum_{g=1}^{c_k}\hat{H}_g + \sqrt{N}\hat{\Omega}\gamma_i\right)\delta\left(A - \phi + \sum_{g=1}^{c_k}\hat{A}_g\right). \tag{C.38}$$

Equations (C.33)–(C.36) constitute functional equations under the constraint (C.37). This constraint corresponds to the normalization constraints in (4.9). By solving these equations, we obtain the distribution of the largest eigenvector elements.

As in the canonical case, solving the functional form of equations is still not analytically tractable. Thus, we again introduce the EMA, i.e., the precision parameters of the Gaussian mixtures A and $\hat{A}$ are fixed as constants, i.e., $q_k(A,H) = q(H)\delta(A-a_k)$ and $\hat{q}_k(\hat{A},\hat{H}) =$

$\hat{q}_k(\hat{H})\delta(\hat{A}-\hat{a}_k)$. Performing the EMA for (C.32), we have

$$
\begin{aligned}
[\lambda(M)]_M = \underset{\phi,\Omega,\hat{\Omega},m_{1k},m_{2k},\hat{m}_{1k},\hat{m}_{2k},a_k,\hat{a}_k}{\mathrm{extr}} \Bigg[& \frac{c_1 p_1^2}{p_1+p_2+\epsilon p_1}\frac{a_1 m_{21}+m_{11}^2}{a_1^2-1} + \frac{c_2\sigma p_2^2}{p_1+\sigma p_2+p_3}\frac{a_2 m_{22}+m_{12}^2}{a_2^2-1} \\
& + \frac{c_3 p_3^2}{p_3+p_2+\epsilon p_3}\frac{a_3 m_{23}+m_{13}^2}{a_3^2-1} + \frac{c_2 p_1 p_2}{p_1+\sigma p_2+p_3}\frac{a_2 m_{21}+a_1 m_{22}+2m_{11}\hat{m}_{11}}{a_1 a_2-1} \\
& + \frac{c_2 p_2 p_3}{p_1+\sigma p_2+p_3}\frac{a_3 m_{22}+a_2 m_{23}+2m_{12}m_{13}}{a_2 a_3-1} + \frac{c_1\epsilon p_1^2}{p_1+p_2+\epsilon p_1}\frac{a_3 m_{21}+a_1 m_{23}+2m_{11}m_{13}}{a_1 a_3-1} \\
& - \sum_k c_k p_k \frac{m_{2k}+2m_{1k}\hat{m}_{1k}+\hat{m}_{2k}}{a_k-\hat{a}_k} + 2\Omega\hat{\Omega}-\Omega^2+\phi \\
& + \frac{1}{N}\sum_k\sum_{i\in V_k}\frac{1}{\phi-c_k\hat{a}_k}\left((\sqrt{N}\hat{\Omega}\gamma_i)^2 - 2\sqrt{N}\hat{\Omega}\gamma_i c_k\hat{m}_{1k} + c_k\hat{m}_{2k} + c_k(c_k-1)\hat{m}_{1k}^2\right)\Bigg],
\end{aligned}
\tag{C.39}
$$

where $m_{\ell k}$ and $\hat{m}_{\ell k}$ represent the ℓth moments of H and $\hat{H}$, respectively, i.e., $m_{\ell k} = \int dH H^\ell q_k(H)$ and $\hat{m}_{\ell k} = \int d\hat{H}\hat{H}^\ell \hat{q}_k(\hat{H})$.

As in the canonical case, we introduce further assumptions. First, we assume the symmetry between the community blocks, namely $p_1 = p_3$ and $c_1 = c_3$. Hence, $a_1 = a_3$, $\hat{a}_1 = \hat{a}_3$, $m_{21} = m_{23}$, and $\hat{m}_{21} = \hat{m}_{23}$. Second, we think of two types of solutions: $m_{11} = -m_{13}$, $m_{12} = 0$ and $m_{11} = m_{12} = m_{13} = 0$. Under these assumptions, we obtain the same solutions as those of the canonical SBM with the regular approximation. When $m_{11} = -m_{13}$ and $m_{12} = 0$, the average largest eigenvalue is obtained as in Eq. (B.36). When $m_{11} = m_{12} = m_{13} = 0$, the average largest eigenvalue is obtained as in Eq. (B.47). The detectability limit is given by Eq. (B.37).

C.3 Saddle-point conditions for $\mathcal{N}_G$

The goal of this subsection is to derive the relations of the normalization constants of the Gaussian mixtures T_k and $\hat{T}_k$ in (C.25) and (C.26). They can be derived using saddle-point conditions for the number of all realizable graphs $\mathcal{N}_G$. This can be calculated by taking the sum over all possible graph configurations as imposing the microcanonical constraints by

delta functions. Thus, we have

$$\begin{aligned}
\mathcal{N}_G = & \sum_{\{u_{ij}\},\{w_{ij}\},\{v_{ij}\},\{w_{ij}\}} \prod_{i\in V_1} \delta\left(\sum_{l\in V_1} u_{il} + \sum_{m\in V_2} v_{im} + \sum_{n\in V_3} w_{in} - c_1\right) \\
& \times \prod_{j\in V_2} \delta\left(\sum_{l\in V_1} u_{jl} + \sum_{m\in V_2} v_{jm} + \sum_{n\in V_3} w_{jn} - c_2\right) \prod_{k\in V_3} \delta\left(\sum_{l\in V_1} u_{kl} + \sum_{m\in V_2} v_{km} + \sum_{n\in V_3} w_{kn} - c_3\right) \\
& \times \delta\left(\sigma p_2 \sum_{i\in V_1}\sum_{j\in V_2} v_{ij} - p_1 \sum_{i,j\in V_2} y_{ij}\right) \delta\left(\sigma p_2 \sum_{i\in V_2}\sum_{j\in V_3} v_{ij} - p_3 \sum_{i,j\in V_2} y_{ij}\right) \\
& \times \delta\left(p_2 \sum_{i,j\in V_1} u_{ij} - p_1 \sum_{i\in V_1}\sum_{j\in V_2} v_{ij}\right) \delta\left(p_2 \sum_{i,j\in V_3} u_{ij} - p_3 \sum_{i\in V_2}\sum_{j\in V_3} v_{ij}\right) \\
& \times \delta\left(\epsilon \sum_{i,j\in V_1} u_{ij} - \sum_{i\in V_1}\sum_{j\in V_3} w_{ij}\right) \delta\left(\epsilon \sum_{i,j\in V_3} u_{ij} - \sum_{i\in V_2}\sum_{j\in V_3} w_{ij}\right). \qquad \text{(C.40)}
\end{aligned}$$

Using the integral representation of the delta function (C.8) and (C.9), we have

$$\begin{aligned}
\mathcal{N}_G = & \sum_{\{u_{ij}\},\{w_{ij}\},\{v_{ij}\},\{w_{ij}\}} \oint \prod_{i\in V_1} \frac{dz_i}{2\pi} z_i^{\sum_{l\in V_1} u_{il} + \sum_{m\in V_2} v_{im} + \sum_{n\in V_3} w_{in} - c_1 - 1} \\
& \times \oint \prod_{i\in V_2} \frac{dz_i}{2\pi} z_i^{\sum_{l\in V_1} v_{il} + \sum_{m\in V_2} y_{im} + \sum_{n\in V_3} v_{in} - c_2 - 1} \oint \prod_{i\in V_3} \frac{dz_i}{2\pi} z_i^{\sum_{l\in V_1} u_{il} + \sum_{m\in V_2} v_{im} + \sum_{n\in V_3} w_{in} - c_3 - 1} \\
& \times \int \frac{d\zeta}{2\pi} e^{-\zeta\left(\sigma p_2 \sum_{i\in V_1}\sum_{j\in V_2} v_{ij} - p_1 \sum_{i,j\in V_2} y_{ij}\right)} \int \frac{d\xi}{2\pi} e^{-\xi\left(\sigma p_2 \sum_{i\in V_2}\sum_{j\in V_3} v_{ij} - p_3 \sum_{i,j\in V_2} y_{ij}\right)} \\
& \times \int \frac{d\tau}{2\pi} e^{-\tau\left(p_2 \sum_{i,j\in V_1} u_{ij} - p_1 \sum_{i\in V_1}\sum_{j\in V_2} v_{ij}\right)} \int \frac{d\kappa}{2\pi} e^{-\kappa\left(p_2 \sum_{i,j\in V_1} u_{ij} - p_3 \sum_{i\in V_2}\sum_{j\in V_3} v_{ij}\right)} \\
& \times \int \frac{d\eta}{2\pi} e^{-\eta\left(\epsilon \sum_{i,j\in V_1} u_{ij} - \sum_{i\in V_1}\sum_{j\in V_3} w_{ij}\right)} \int \frac{d\theta}{2\pi} e^{-\theta\left(\epsilon \sum_{i,j\in V_3} u_{ij} - \sum_{i\in V_1}\sum_{j\in V_3} w_{ij}\right)} \qquad \text{(C.41)}
\end{aligned}$$

$$\begin{aligned}
= & \oint \prod_{k=1,2,3} \prod_{i\in V_k} \frac{dz_i}{2\pi} z_i^{-(1+c_k)} \int \frac{d\zeta}{2\pi} \int \frac{d\xi}{2\pi} \int \frac{d\tau}{2\pi} \int \frac{d\kappa}{2\pi} \int \frac{d\eta}{2\pi} \int \frac{d\theta}{2\pi} \\
& \times \prod_{\substack{i<j \\ i,j\in V_1}} \sum_{u_{ij}} \left(z_i z_j e^{-2\tau p_2 - 2\eta\epsilon}\right)^{u_{ij}} \prod_{\substack{i<j \\ i,j\in V_2}} \sum_{y_{ij}} \left(z_i z_j e^{2\xi p_3 + 2\zeta p_1}\right)^{y_{ij}} \prod_{\substack{i<j \\ i,j\in V_3}} \sum_{u_{ij}} \left(z_i z_j e^{-2\kappa p_2 - 2\theta\epsilon}\right)^{u_{ij}} \\
& \times \prod_{i\in V_1}\prod_{j\in V_2} \sum_{v_{ij}} \left(z_i z_j e^{-\sigma\zeta p_2 + \tau p_1}\right)^{v_{ij}} \prod_{i\in V_2}\prod_{j\in V_3} \sum_{v_{ij}} \left(z_i z_j e^{-\sigma\xi p_2 + \kappa p_3}\right)^{v_{ij}} \prod_{i\in V_1}\prod_{j\in V_3} \sum_{w_{ij}} \left(z_i z_j e^{\eta+\theta}\right)^{w_{ij}}. \qquad \text{(C.42)}
\end{aligned}$$

Here, we introduce the order parameters

$$q_k = \frac{1}{p_k N} \sum_{i\in V_k} z_i. \quad (k = 1, 2, 3) \qquad \text{(C.43)}$$

Equation (C.42) is now written as

$$\mathcal{N}_G = \prod_{k=1,2,3}\left(p_k N\int dq_k \prod_{i\in V_k}\oint \frac{dz_i}{2\pi} z_i^{-(1+c_k)}\right)\int\frac{d\zeta}{2\pi}\int\frac{d\xi}{2\pi}\int\frac{d\tau}{2\pi}\int\frac{d\kappa}{2\pi}\int\frac{d\eta}{2\pi}\int\frac{d\theta}{2\pi}$$
$$\times\prod_{k=1,2,3}\delta\left(p_k N q_k - \sum_{i\in V_k} z_i\right)\exp\left(\frac{1}{2}e^{-2\tau p_2-2\epsilon\eta}(p_1Nq_1)^2+\frac{1}{2}e^{2\zeta p_1+2\xi p_3}(p_2Nq_2)^2\right.$$
$$\left.+\frac{1}{2}e^{-2\kappa p_2-2\xi\theta}(p_3Nq_3)^2+e^{-\sigma\zeta p_2+\tau p_1}p_1p_2N^2q_1q_2+e^{-\sigma\xi p_2+\kappa p_3}p_2p_3N^2q_2q_3+e^{\eta+\theta}p_1p_3N^2q_1q_3)\right). \tag{C.44}$$

Here, we used the same approximation as in (C.11). Using relations (C.9) and (C.24), Eq. (C.44) becomes

$$\mathcal{N}_G = \prod_{k=1,2,3}\left(p_k N\int\frac{dq_k d\hat{q}_k}{2\pi}\right)\int\frac{d\zeta}{2\pi}\int\frac{d\xi}{2\pi}\int\frac{d\tau}{2\pi}\int\frac{d\kappa}{2\pi}\int\frac{d\eta}{2\pi}\int\frac{d\theta}{2\pi}$$
$$\times\exp\left(\frac{1}{2}e^{-2\tau p_2-2\epsilon\eta}(p_1Nq_1)^2+\frac{1}{2}e^{2\zeta p_1+2\xi p_3}(p_2Nq_2)^2\right.$$
$$+\frac{1}{2}e^{-2\kappa p_2-2\xi\theta}(p_3Nq_3)^2+e^{-\sigma\zeta p_2+\tau p_1}p_1p_2N^2q_1q_2$$
$$+e^{-\sigma\xi p_2+\kappa p_3}p_2p_3N^2q_2q_3+e^{\eta+\theta}p_1p_3N^2q_1q_3)$$
$$\left.+N\sum_{k=1,2,3}(-\hat{q}_k p_k q_k+p_k c_k\log\hat{q}_k-p_k\log c_k!)\right). \tag{C.45}$$

In the limit $N\to\infty$, we have the following saddle-point conditions.

$$\epsilon p_1q_1e^{-2\tau p_2-2\epsilon\eta} = p_3q_3e^{\eta+\theta} \tag{C.46}$$
$$\epsilon p_3q_3e^{-2\kappa p_2-2\epsilon\theta} = p_1q_1e^{\eta+\theta} \tag{C.47}$$
$$q_2e^{2\zeta p_1+2\xi p_3} = \sigma q_1e^{-\sigma\zeta p_2+\tau p_1} \tag{C.48}$$
$$q_1e^{-2\tau p_2-2\epsilon\eta} = q_2e^{-\sigma\zeta p_2+\tau p_1} \tag{C.49}$$
$$q_3e^{-2\kappa p_2-2\epsilon\theta} = q_2e^{-\sigma\xi p_2+\kappa p_3} \tag{C.50}$$

$$\frac{\hat{q}_1}{N} = p_1q_1e^{-2\tau p_2-2\epsilon\eta}+p_2q_2e^{-\sigma\zeta p_2+\tau p_1}+p_3q_3e^{\eta+\theta} \tag{C.51}$$
$$\frac{\hat{q}_2}{N} = p_2q_2e^{-2\zeta p_1+2\xi p_3}+p_1q_1e^{-\sigma\zeta p_2+\tau p_1}+p_3q_3e^{-\sigma\xi p_2+\kappa p_3} \tag{C.52}$$
$$\frac{\hat{q}_3}{N} = p_3q_3e^{-2\kappa p_2-2\epsilon\theta}+p_2q_2e^{-\sigma\xi p_2+\kappa p_3}+p_1q_1e^{\eta+\theta} \tag{C.53}$$
$$q_k\hat{q}_k = c_k.\quad(k=1,2,3) \tag{C.54}$$

From Eq. (C.46)–(C.54), we obtain

$$q_1^2 = \frac{1}{N} e^{2\tau p_2 + 2\epsilon\eta} \frac{c_1}{p_1 + p_2 + \epsilon p_1}, \tag{C.55}$$

$$q_2^2 = \frac{1}{N} e^{-2\zeta p_1 - 2\xi p_3} \frac{c_2 \sigma}{p_1 + \sigma p_2 + p_3}, \tag{C.56}$$

$$q_3^2 = \frac{1}{N} e^{2\kappa p_2 + 2\epsilon\theta} \frac{c_3}{p_3 + p_2 + \epsilon p_3}, \tag{C.57}$$

$$q_1 q_2 = \frac{1}{N} e^{\sigma\zeta p_2 - \tau p_1} \frac{c_2}{\sigma p_2 + p_1 + p_3}, \tag{C.58}$$

$$q_2 q_3 = \frac{1}{N} e^{\sigma\xi p_2 - \kappa p_3} \frac{c_2}{\sigma p_2 + p_1 + p_3}, \tag{C.59}$$

$$q_1 q_3 = \frac{1}{N} e^{-(\eta+\theta)} \frac{p_1}{p_3} \frac{c_1 \epsilon}{p_1 + p_2 + \epsilon p_1}, \tag{C.60}$$

$$c_1(\sigma p_2 + p_1 + p_3) = c_2(p_1 + p_2 + \epsilon p_3). \tag{C.61}$$

By substituting (C.55)–(C.61) into (C.45), $\mathcal{N}_G$ is expressed in terms of the model parameters. The order parameters (C.43) correspond to the order-parameter functions (B.10) when $n = 0$. This indicates that the normalization constants of the Gaussian mixtures T_k and $\hat{T}_k$ in (C.25) and (C.26) are identical to q_k and $\hat{q}_k$, respectively. Accordingly, we obtain the relations between T_k and $\hat{T}_k$ as Eqs. (C.54)–(C.60). Besides, (C.61) is identical to the constraint between the model parameters (4.6), i.e., the same constraint is derived by both the model definition and the replica analysis.

C.4 Comparison between the canonical and microcanonical SBMs

In the main text, we used the canonical SBM for deriving the detectability limit, whereas we used the microcanonical SBM for conducting the numerical experiments. This is because the derivation under the canonical SBM is more straightforward and simpler, while the canonical SBM causes a problem when conducting the numerical experiments. The canonical SBM required the regular approximation as an additional approximation to calculate the average largest eigenvalue in the replica analysis. The approximation creates a large difference of the derived solutions from the original ones because of ignoring the fluctuation of the degree distribution. Thus, it becomes difficult to validate the results of the analytical calculation by comparing them to the results of the numerical experiments.

However, the microcanonical SBM does not require the regular approximation because it can be defined with an arbitrary degree sequence, and we can choose one that avoids the effects of the fluctuation. Meanwhile, as mentioned in Sec. 4.3.1, the microcanonical SBM requires an additional constraint that c_1 and c_2 can take only natural numbers. This originates from the fact that it specifies a certain degree for each node as its model parameters. Note that the replica analysis with the microcanonical SBM (and canonical SBM) required

another approximation, which is called EMA. However, the effect of this approximation can be neglected under the experimental condition in Sec. 4.3, as discussed in Appendix E.

In short, the canonical SBM is appropriate to explain the derivation of the detectability limit because of the simplicity. The microcanonical SBM is appropriate for conducting the numerical experiments because it does not require the regular approximation.

Appendix D

Bimodal stochastic block model

In this appendix, we explain the bimodal SBM in detail. This model is a variant of the SBM that has no overlapping structure. The bimodal SBM has a bimodal degree distribution: each node randomly takes either degree c_1 or c_2. We denote the fraction of the nodes that have degree c_1 as b_1 and that of c_2 as b_2 ($b_1 + b_2 = 1$). Note that, because the degree assignment is independent of the group assignment, one cannot infer the planted structure based on the degree sequence.

We define the two-blcok bimodal SBM in the microcanonical formulation. The model is parametrized as follows.

$$\boldsymbol{e} = \begin{pmatrix} 1 & \epsilon \\ \epsilon & 1 \end{pmatrix} e_{11}, \tag{D.1}$$

$$\boldsymbol{b} = (b_1, b_2) = (2p_1, p_2). \tag{D.2}$$

Here, as defined in Sec. 4.1, e_{kl} is the number of edges between blocks k and l, and ϵ is the parameter that controls the strength of community structure. Moreover, p_1 and $p_2 (= \alpha p_1)$ are the sizes of the community and overlapping blocks of the overlapping SBM, respectively. As mentioned in the main text, the purpose of introducing the bimodal SBM is to compare the overlapping SBM to the SBM with the non-overlapping structure and the same average degree. We can confirm that both models have the same average degree.

Subsequently, we show the average largest eigenvalue of the bimodal SBM under the detectable and undetectable conditions. As in the overlapping SBM, we can calculate it using the replica method. The detailed derivation can be found in Ref. [68].

First, under the detectable condition, we obtain the equation for a as

$$\overline{c_b}(c_2 A - B)(c_1 A - B) = (a^2 - 1)\left(\overline{c_b} A - B\right) B, \tag{D.3}$$

where

$$A = (\overline{c_b} - 1)\Gamma - a, \tag{D.4}$$

$$B = \Gamma(\overline{c^2} - \overline{c_b}) - a\overline{c_b}, \tag{D.5}$$

$$\Gamma = \frac{1 - \epsilon}{1 + \epsilon}. \tag{D.6}$$

Here, a is the precision parameter of the Gaussian mixture, which corresponds to a_1 and a_2 in the case of the overlapping SBM. Besides, $\overline{c_b} \equiv b_1 c_1 + b_2 c_2$ and $\overline{c_b^2} \equiv b_1 c_1^2 + b_2 c_2^2$. Note that a has no indices because of the symmetry between the two blocks. We let the solutions of Eq. (D.3) be a^{det}. Using this solution, we obtain the following expression of the average largest eigenvalue.

$$[\lambda(M)]_M = \frac{c_1 c_2}{(a^{\mathrm{det}})^3} \frac{A}{B}. \tag{D.7}$$

Second, under the undetectable case, we obtain the equations for a and ϕ as follows.

$$\sum_{t=1,2} \frac{b_t c_t^2}{(\phi - c_t/a)^2} = \frac{(a^2+1)}{\overline{c_b}} \left(\frac{\overline{c_b} a}{a^2 - 1} \right)^2. \tag{D.8}$$

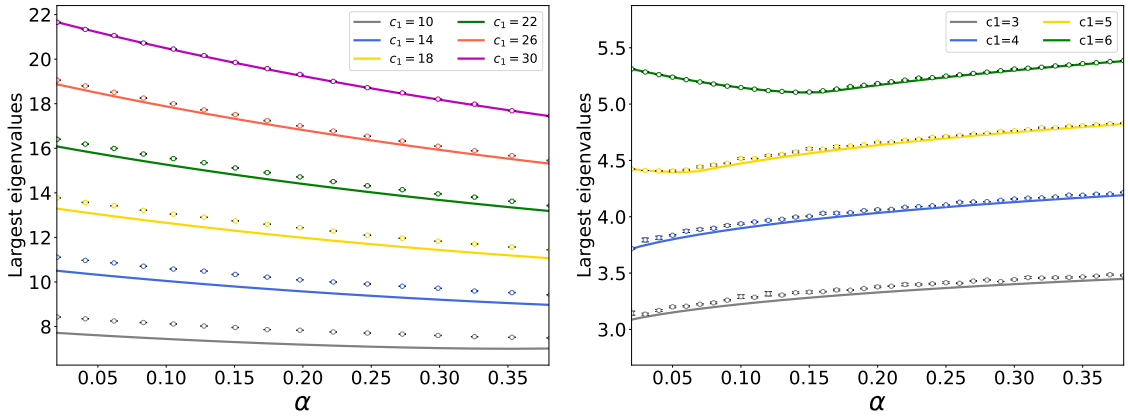
When we let the solutions of these equations be a^{und} and ϕ^{und}, we obtain the average largest eigenvalue as $[\lambda(M)]_M = \phi^{\mathrm{und}}$.

Appendix E

Accuracies of the EMA and the regular approximation

For the replica analysis, we introduced two approximations: the regular approximation and EMA. Here, we investigate the dependencies of the average degree on the accuracy of each approximation. It is known that when the average degree is sufficiently large, the effect of these approximations can be asymptotically ignored. However, it is not trivial how the approximations affect the results for a graph with a low average degree.

To derive the detectability limit of the canonical SBM, we used both the EMA and the regular approximation. To derive that of the microcanonical SBM, we used the EMA only. Thus, by comparing both results, we can measure how each approximation differs from the original result. Figs. E.1a and E.1b show the results of the canonical and microcanonical SBMs, respectively. We can see that the results of the replica analysis and the numerical experiments are in agreement for $c_1 \geq 30$ in the canonical case. On the other hand, they are in agreement for $c_1 \geq 6$ in the microcanonical case. Therefore, we can conclude that the effect of the EMA is smaller than that of the regular approximation. Therefore, for the numerical experiments in Sec. 4.3, we used the microcanonical SBM and set $c_1 = 10$, so that the effect of the approximation can be ignored.



(a) (b)

Figure E.1: Largest eigenvalues as a function of α. The lines represent the results of the replica analysis and the dots represent those of the numerical experiments. (a) The figure shows the results of the canonical SBM for $c_1 = 10, 14, 18, 22, 26, 30$. (b) The figure shows the results of the microcanonical SBM for $c_1 = 3, 4, 5, 6$. The figures are taken from Ref. [88].

Appendix F

Relationship with the mixed-membership SBM

Mixed-membership stochastic block model (MMSBM) [20] is a popular random graph model that considers an overlapping structure. In this section, we discuss the relationship between our overlapping SBM and the MMSBM. We define a membership vector of node i as $\boldsymbol{\pi}_i = [\pi_{ik}]$ ($k \in \{1, \dots, K\}$), $\sum_{k=1}^{K} \pi_{ik} = 1$, $0 < \pi_{ik} \leq 1$. That is, π_{ik} represents the probability that node i is assigned to block k. In the MMSBM, the edge generation probability of a pair of nodes (i, j) is expressed as

$$P(A_{ij} = 1 | \boldsymbol{\rho}, \boldsymbol{\pi}_i, \boldsymbol{\pi}_j) = \boldsymbol{\pi}_i^\top \boldsymbol{\rho} \boldsymbol{\pi}_j. \tag{F.1}$$

To see the correspondence to our overlapping SBM, we consider a two-block MMSBM, and exclusive node sets V_1, V_2, and V_3, where V_1 and V_3 represent community blocks and V_2 represents the overlapping block. For example, let us consider the following parameterization of $\boldsymbol{\pi}_i$.

$$\boldsymbol{\pi}_i = \begin{cases} (1,0)^\top & (i \in V_1) \\ (1/2, 1/2)^\top & (i \in V_2) \\ (0,1)^\top & (i \in V_3). \end{cases} \tag{F.2}$$

We consider the same parameterization as Eq. (4.3) for the affinity matrix $\boldsymbol{\rho}$. By inserting Eqs. (4.3) and (F.2) into Eq. (F.1), we obtain the edge generation probability matrix

$$\begin{pmatrix} \rho_{\mathrm{in}} & \frac{\rho_{\mathrm{in}}+\rho_{\mathrm{out}}}{2} & \rho_{\mathrm{out}} \\ \frac{\rho_{\mathrm{in}}+\rho_{\mathrm{out}}}{2} & \frac{\rho_{\mathrm{in}}+\rho_{\mathrm{out}}}{2} & \frac{\rho_{\mathrm{in}}+\rho_{\mathrm{out}}}{2} \\ \rho_{\mathrm{out}} & \frac{\rho_{\mathrm{in}}+\rho_{\mathrm{out}}}{2} & \rho_{\mathrm{in}} \end{pmatrix}. \tag{F.3}$$

This equation never coincides with Eq. (4.5). In fact, one can easily confirm that the MMSBM does not coincide with our overlapping SBM for arbitrary choices of $\boldsymbol{\pi}_i$ in Eq. (F.2).

It is interesting to consider a variant of the standard MMSBM. We define a membership vector of node i as an unnormalized propensity vector $\boldsymbol{g}_i = [g_{ik}]$ ($k \in \{1, \dots, K\}$), $g_{ik} \geq 0$. Similarly to the standard MMSBM, the edge generation probability of a pair of nodes (i, j) is expressed as

$$P(A_{ij} = 1 | \boldsymbol{\rho}, \boldsymbol{g}_i, \boldsymbol{g}_j) = \boldsymbol{g}_i^\top \boldsymbol{\rho} \boldsymbol{g}_j. \tag{F.4}$$

Again, we consider the case of $K = 2$ and the following parameterization of $\boldsymbol{g}_i$.

$$\boldsymbol{g}_i = \begin{cases} (1,0)^\top & (i \in V_1) \\ (\frac{1}{1+\epsilon}, \frac{1}{1+\epsilon})^\top & (i \in V_2) \\ (0,1)^\top & (i \in V_3). \end{cases} \tag{F.5}$$

Here, the labels of the two community blocks are exchangeable because of the permutation symmetry. By inserting Eqs. (4.3) and (F.5) into Eq. (F.4), we obtain the edge generation probability matrix

$$\begin{pmatrix} \rho_{\mathrm{in}} & \rho_{\mathrm{in}} & \rho_{\mathrm{out}} \\ \rho_{\mathrm{in}} & \frac{2}{1+\epsilon}\rho_{\mathrm{in}} & \rho_{\mathrm{in}} \\ \rho_{\mathrm{out}} & \rho_{\mathrm{in}} & \rho_{\mathrm{in}} \end{pmatrix}. \tag{F.6}$$

Equation (F.6) becomes identical to Eq. (4.5) when $\sigma = 2/(1+\epsilon)$. In fact, one can confirm that the parameterization of $\boldsymbol{\pi}_i$ in Eq. (F.5) is the only nontrivial choice that achieves the equivalence to our overlapping SBM. Therefore, this generalized MMSBM and our overlapping SBM share the same model space in the range of $1 \leq \sigma \leq 2$.