Eötvös Loránd University
Faculty of Science

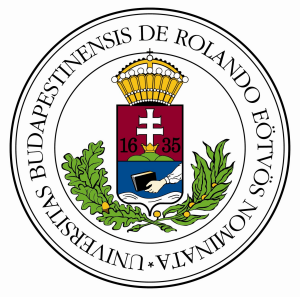
Sebestyén Kovács

Random Matrices, Perturbations and Their Applications in Statistics

MSc Thesis
Applied Mathematics

Supervisor:

Ágnes Backhausz, assistant professor
Department of Probability Theory and Statistics



Budapest, 2025.

ACKNOWLEDGEMENTS

I would like to express my heartfelt gratitude to Dr. Ágnes Backhausz for her dedicated guidance and support in supervising my thesis. I truly appreciated the opportunity to work on the problems she proposed, and I found great joy in solving them throughout the process.

I am deeply thankful to my parents and brothers for their unwavering support and encouragement. Their presence and belief in me have been a constant source of strength during my studies.

Budapest, May 30, 2025.

Sebestyén Kovács

Contents

Introduction

In recent years, the study of random matrices and their perturbations has become increasingly relevant in statistics, data science and network theory. These tools offer powerful methods for uncovering hidden structures in complex datasets, especially when data are represented in matrix form, such as in adjacency matrices of graphs or covariance matrices in multivariate statistics.

This thesis is centered around the interplay between random matrix theory and statistical applications, with a special focus on the behavior of singular and eigenvectors under random perturbations. In the first part, we examine how singular vectors of low-rank matrices are affected by the addition of noise. This analysis is crucial for understanding the robustness of spectral methods and for developing algorithms that remain reliable under uncertainty.

The second part of the thesis focuses on the Stochastic Block Model (SBM), a widely studied probabilistic model for community detection in networks. We investigate both theoretical aspects and practical implementations of the SBM, including its relation to the $\mathbb{Z}_2$ synchronization problem, as well as the performance of spectral methods in noisy environments. Through simulations, we evaluate misclassification rates and explore how well the underlying structure of a graph can be recovered when observations are corrupted by noise.

In the final chapter, we extend the SBM framework by considering graph powers, which allow us to better capture community structure in sparse graphs. We analyze the spectral properties of the corresponding matrices and show how these techniques lead to improved classification even under adversarial perturbations.

By combining tools from linear algebra, probability theory and statistical inference, this thesis aims to contribute to the understanding of when and how spectral algorithms succeed in reconstructing latent structures from noisy and high-dimensional data.

Chapter 1

Behavior of Singular Vectors of Random Matrices under Perturbation Effects

The study of singular vectors and singular values of random matrices plays a crucial role in understanding complex data structures. In many applications, data can be represented as matrices, where rows correspond to observations and columns to different features. The *Singular Value Decomposition* (SVD) is a key mathematical tool that decomposes a matrix into singular values and singular vectors, enabling us to extract meaningful patterns. This decomposition allows us to better understand data variability and dependencies, revealing significant properties of the dataset. The singular values and singular vectors of a matrix have several crucial applications in data analysis:

- **Dimensionality Reduction**: The leading singular vectors capture the most significant directions of variation, which is essential in techniques such as Principal Component Analysis (PCA).
- **Noise Filtering**: Retaining only the dominant singular values enables the removal of noise while preserving the essential information in the data (cf. [9]).
- **Feature Extraction and Interpretation**: In high-dimensional datasets, singular vectors often correspond to meaningful structures, which aids in machine learning and pattern recognition (cf. [14]).
- **Understanding Covariance Structure**: In statistics, the singular vectors of covariance matrices reveal key dependencies among variables, playing a crucial role in applications ranging from finance to physics and biological sciences.
- **Stability of Solutions**: Many optimization problems involve matrices, and their singular vectors determine the stability and sensitivity of numerical solutions (cf. [19]).

1.1 Singular vectors and values of matrices

By analyzing singular vectors and values, we gain a deeper understanding of data structure, covariance properties and the relationships between variables. This knowledge is essential for effective statistical modeling, machine learning applications and signal processing. In real-world scenarios, data is often subject to measurement errors, missing values or external disturbances. These perturbations can significantly impact the singular values, singular vectors and overall structure of the data matrix. Therefore, studying how small changes affect matrix properties is essential. Key aspects of perturbation analysis include:

- **Sensitivity of Singular Values and Vectors**: Small perturbations in data can lead to significant variations in singular values and singular vectors. Understanding these effects helps in designing robust algorithms (cf. [18]).
- **Low-Rank Approximations in Noisy Data**: Many applications rely on approximating a matrix with a low-rank version (e.g., truncated SVD) to reduce noise and enhance interpretability (cf. [11]).
- **Stability of Covariance Matrices**: Covariance matrices computed from noisy data may be unstable. Perturbation analysis helps determine when eigenvalue shifts lead to unreliable conclusions (cf. [15]).
- **Applications in Machine Learning and Signal Processing**: Many machine learning algorithms depend on singular values and vectors. Studying perturbations allows us to assess algorithm robustness in noisy environments (cf. [20]).

Understanding these aspects ensures that matrix decompositions remain useful even in the presence of uncertainty, making perturbation analysis a critical tool in modern data-driven applications. After this, we can define the singular values and singular vectors of matrices.
Let $\|\cdot\|$ denote the Euclidean norm in $\mathbb{R}^n$.

Definition 1.1 Let σ_1 be the first singular value of the matrix A, i.e.

$$\sigma_1 := \max_{\|v\|=1} \|Av\|,$$

and let denote the first singular vector of the matrix A by v_1:

$$v_1 := \operatorname*{argmax}_{\|v\|=1} \|Av\|.$$

Let σ_2 be the second singular value of the matrix A, i.e.

$$\sigma_2 := \max_{v \perp v_1} \|Av\|,$$

and let denote the second singular vector of the matrix A by v_2:

$$v_2 = \operatorname*{argmax}_{v \perp v_1} \|Av\|.$$

By induction let v_i and σ_i be the i-th singular vector and singular value of A (for $i = 3,4,\ldots,r$), i.e.

$$\sigma_i := \max_{v \perp v_1, v_2, \ldots, v_{i-1}} \|Av\| > 0 \qquad \text{and} \qquad v_i := \operatorname*{argmax}_{v \perp v_1, v_2 \ldots, v_{i-1}} \|Av\|.$$

In this situation we have

$$\max_{v \perp v_1, v_2, \ldots, v_r} \|Av\| = 0.$$

After this definition we will be able to see that for each $1 \leq k \leq r$ the space V_k is the best-fit k-dimensional subspace for the rows of A matrix, where V_k is the subspace spanned by $v_1, v_2, \ldots v_k$.

Theorem 1.2 (cf. [17]) For every $A \in \mathbb{R}^{m \times n}$ the space V_k is the best-fit k-dimensional subspace for the rows of A.

Proof.

By definition, the singular vectors are orthogonal to each other and have unit norms. Now, we seek a subspace of $\mathbb{R}^n$ that is closest to the row vectors of A. Thus, we aim to minimize the following quantity:

$$\min_{W} \sum_{j=1}^{k} d(a_j, W).$$

Here W is a subspace of $\mathbb{R}^n$, and a_i denotes the i-th row of the matrix A.

Let $w_1, w_2, \ldots, w_k$ be an orthonormal basis of an arbitrary subspace W, which exists. By the Pythagorean theorem, the distance between a_i and W is minimized if the norm of the orthogonal projection of a_i onto W is maximized. We know that $w_1, w_2, \ldots, w_k$ forms an orthonormal basis of W, so the norm of the orthogonal projection of a_i onto W is given by

$$\sum_{j=1}^{k} \langle w_j, a_i \rangle^2.$$

Thus, we wish to maximize this function:

$$\max_{w_1, w_2, \ldots, w_k} \sum_{j=1}^{k} \sum_{i=1}^{m} \langle w_j, a_i \rangle^2 = \max_{w_1, w_2, \ldots, w_k} \sum_{j=1}^{k} \|Aw_j\|^2.$$

We assert that the minimum occurs at V_k. We will prove this assertion by mathematical induction.

For $k = 1$, the result is trivial due to the definition of the first singular vector. Now, assume that the best-fit $(k-1)$-dimensional subspace of A is V_{k-1}. Thus, for any subspace W with an orthonormal basis $w_1, w_2, \ldots, w_{k-1}$, we have

$$\sum_{j=1}^{k-1} \|Aw_j\|^2 \leq \sum_{j=1}^{k-1} \|Av_j\|^2.$$

We can assume that w_k is orthogonal to the vectors $v_1, v_2, \ldots, v_{k-1}$. Therefore, by the definition of the singular vectors,

$$\|Aw_k\|^2 \leq \|Av_k\|^2.$$

By adding the two inequalities, we obtain the proof of the theorem.

From now on, we focus on the problem of how the singular vectors change when a random noise matrix E is added to A. First, we examine the case when the noise matrix is a Bernoulli matrix, i.e.

Definition 1.3 An E matrix is called a Bernoulli matrix if its components are independent and each component is a random sign different from zero, that is,

$$E = [E]_{i,j}, \qquad P(E_{i,j} = 1) = P(E_{i,j} = -1) = \frac{1}{2}.$$

If a matrix has large singular values, it indicates that the matrix performs strong transformations and is well-conditioned. This means that the matrix can stretch or compress data significantly, and its inverse (if it exists) is stable. On the other hand, if a matrix has small singular values, it

suggests that the matrix is poorly conditioned, meaning it may cause instability or numerical problems when performing transformations. Small singular values indicate that the matrix compresses data in certain directions, and its inverse (if it exists) may be sensitive to small perturbations in the input. The following theorem states that if the first singular value of a matrix is sufficiently large, and a Bernoulli noise matrix is added to it, the first singular vector of the resulting new matrix will be close to the first singular vector of the original matrix:

Theorem 1.4 (cf. [17]) Assume that E is a Bernoulli matrix and $A,E \in \mathbb{R}^{n\times n}$, furthermore let the rank of A be denoted by r. For every $\varepsilon > 0$ there exist constants $C,\delta_0 > 0$ such that if

$$\delta \geq \delta_0 \quad \text{and} \quad \sigma_1 \geq \max\{n,\sqrt{n}\cdot\delta\}$$

then with a probability at least $1-\varepsilon$ the inequality

$$\sin(< (v_1,v_1')) \leq C\cdot\frac{\sqrt{r}}{\delta}$$

fulfils. Here v_1 is the first singular vector of matrix A and v_1' is the first singular vector of $A+E$ (the new matrix).

It is evident that the two vectors in question have unit length, and if the conditions of the theorem are satisfied, they form a small angle. Therefore, the first singular vector describing the system does not change in our case. (The proofs are omitted in the original paper.) According to the article by O'Rourke, Wang, and Vu, two lemmas played a crucial role in the proof of this theorem. By the way, although the proof of the first lemma was only superficially presented in the article, we developed the essential part of the proof independently. Regarding the second lemma, no proof was provided by the authors; however, after understanding the first, we were able to construct a proof ourselves. Below, we present the proofs of these two lemmas.

Lemma 1.5 (cf. [17]) Let $E = (\xi_{ij})_{i,j=1}^{n}$ be an $n\times n$ real symmetric random matrix, where $\{\xi_{ij} : 1 \leq i \leq j \leq n\}$ is a collection of independent random variables, each with mean zero. Further assume that

$$\sup_{1\leq i\leq j\leq n} |\xi_{ij}| \leq K$$

with probability 1, for some $K \geq 1$. Then for any fixed unit vectors u,v and every $t > 0$,

$$P(|u^T E v| \geq t) \leq 2\cdot\exp\left(-\frac{t^2}{8K^2}\right).$$

Proof. In the proof, we will apply the Azuma-Hoeffding inequality, which states the following:
Let $X_1, \ldots, X_n$ be independent random variables where, for each i, X_i takes values in $[a_i,b_i]$ with $-\infty < a_i \leq b_i < +\infty$. Let

$$S_n := \sum_{i=1}^{n} X_i.$$

Then for all $t > 0$ we have

$$P(|S_n - \mathbb{E}[S_n]| \geq t) \leq 2 \cdot \exp\left(-\frac{2t^2}{\sum_{i=1}^{n}(b_i - a_i)^2}\right).$$

It is evident that

$$u^T E v = \sum_{i=1}^{n}\left(\sum_{j=1}^{n} \xi_{ij} u_j\right) v_i = \sum_{i=1}^{n} \xi_{ii} u_i v_i + \sum_{i<j} \xi_{ij}\left(u_i v_j + u_j v_i\right).$$

We define $X_1, X_2, \ldots X_n$ as follows:

$$\begin{aligned} X_1 &:= u_1 v_1 \xi_{11} + \sum_{j=2}^{n} \xi_{1j}(u_1 v_j + u_j v_1), \\ &\vdots \\ X_n &:= u_n v_n \xi_{nn} + \sum_{j=1}^{n-1} \xi_{nj}(u_n v_j + u_j v_n). \end{aligned}$$

Then $X_1, X_2, \ldots, X_n$ are independent random variables, and

$$S_n = \sum_{i=1}^{n} X_i = u^T E v.$$

Now for $i = 1, \ldots, n$

$$\mathbb{E}(X_i) = 0 \quad \Rightarrow \quad \mathbb{E}(S_n) = 0.$$

In this case

$$\begin{aligned} |X_1| &\leq \left(|u_1 v_1| + \sum_{j=2}^{n} |u_1 v_j| + |u_j v_1|\right) K =: b_1, \qquad a_1 := -b_1, \\ &\vdots \\ |X_n| &\leq \left(|u_n v_n| + \sum_{j=1}^{n-1} |u_n v_j| + |u_j v_n|\right) K =: b_n, \qquad a_n := -b_n. \end{aligned}$$

We can use the following:

$$(b_1-a_1)^2+\ldots+(b_n-a_n)^2=4(b_1^2+\cdots+b_n^2)$$

$$= 4K^2\left(\sum_{i<j}|u_iv_j|^2+|v_iu_j|^2+2\sum_{i<j}|u_iv_ju_jv_i|+\sum_{i=1}^{n}|u_iv_i|^2\right)$$

$$\leq 4K^2\cdot 4\left(\sum_{i=1}^{n}\sum_{j=1}^{n}|u_iv_j|^2\right)=16K^2\left(\sum_{i=1}^{n}u_i^2\right)\left(\sum_{j=1}^{n}v_j^2\right)=16K^2\cdot 1\cdot 1=16K^2.$$

Therefore

$$P(\left|u^TEv\right|\geq t)=P(|S_n-\mathbb{E}[S_n]|\geq t)\leq 2\cdot\exp\left(-\frac{2t^2}{16K^2}\right)=2\cdot\exp\left(-\frac{t^2}{8K^2}\right).$$

The second lemma is very similar to the first one, but here the random matrix in question is not necessarily symmetric.

Lemma 1.6 (cf. [17]) Let $E=(\xi_{ij})_{1\leq i\leq m,1\leq j\leq n}$ be an $m\times n$ real random matrix, where $\{\xi_{ij}:1\leq i\leq m,1\leq j\leq n\}$ is a collection of independent random variables, each with mean zero. Furthermore, assume that for some $K\geq 1$,

$$\sup_{1\leq i\leq m,1\leq j\leq n}|\xi_{ij}|\leq K$$

holds with probability 1. Then, for any fixed unit vectors $u\in\mathbb{R}^m$ and $v\in\mathbb{R}^n$, and for every $t>0$, we have

$$P(|u^TEv|\geq t)\leq 2\exp\left(-\frac{t^2}{2K^2}\right). \tag{1.1.1}$$

Proof. The proof is also very similar to the proof of the previous lemma, so we only present the essential changes. Due to the definition of the energy inner product

$$u^TEv=\sum_{j=1}^{n}\sum_{i=1}^{m}\xi_{ij}u_iv_j.$$

We can express this in summation form:

$$\begin{aligned}X_1 &:= \sum_{i=1}^{m}\xi_{i1}u_iv_1\\ &\vdots\\ X_n &:= \sum_{i=1}^{m}\xi_{in}u_iv_n.\end{aligned}$$

Similarly to the previous cases:

$$\begin{aligned} |X_1| &\leq K \cdot \textstyle\sum_{i=1}^{m} |u_i v_1| =: b_1, \qquad a_1 := -b_1, \\ &\vdots \\ |X_n| &\leq K \cdot \textstyle\sum_{i=1}^{m} |u_i v_n| =: b_n, \qquad a_n := -b_n, \end{aligned}$$

Our previous estimate is modified as follows:

$$\sum_{j=1}^{n} (b_j - a_j)^2 = 4 \cdot \sum_{j=1}^{n} b_j^2 = 4K^2 \sum_{j=1}^{n} \left(\sum_{i=1}^{m} |u_i v_j| \right)^2 = 4K^2 \left(\sum_{j=1}^{n} v_j^2 \right) \cdot \left(\sum_{i=1}^{m} u_i^2 \right)$$

$$= \quad 4K^2 \cdot 1 \cdot 1 = 4K^2.$$

And finally, using the Azuma-Hoeffding inequality

$$P(\left|u^T E v\right| \geq t) = P(|S_n - \mathbb{E}[S_n]| \geq t) \leq 2 \cdot \exp\left(-\frac{2t^2}{4K^2}\right) = 2 \cdot \exp\left(-\frac{t^2}{2K^2}\right).$$

Note. Lemma 1.5 and Lemma 1.6 mean that if the energy scalar products of unit vectors are associated with a random matrix having independent, zero mean and bounded components, then these scalar products are unlikely to take large values with high probability.

1.2 Singular Value Decomposition (SVD)

Singular Value Decomposition is essential for revealing the underlying structure of complex datasets. It simplifies data by breaking it down into its most significant components, making it easier to identify patterns. SVD is crucial for reducing dimensionality, which helps improve computational efficiency without losing critical information. It plays a vital role in extracting features from large datasets, enhancing the interpretability of machine learning models. Additionally, SVD aids in identifying relationships between variables, contributing to better statistical analysis and data-driven decision-making. Using singular value decomposition, we can redefine the singular values and singular vectors of matrices.

Definition 1.7 The Singular Value Decomposition (SVD) says if $A \in \mathbb{R}^{d_1 \times d_2}$, $r(A) = n$ then there exists only one $U \in \mathbb{R}^{d_1 \times n}$, $\Sigma \in \mathbb{R}^{n \times n}$ and $V \in R^{d_2 \times n}$ such that

$$A = U\Sigma V^T = \sum_{i=1}^{n} \sigma_i u_i v_i^T,$$

where U and V matrices are orthogonal with $u_1, u_2, \ldots, u_n$ and $v_1, v_2, \ldots, v_n$ column vectors and Σ is a diagonal matrix with

$$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n > 0$$

in the main diagonal. We say that v_i and σ_i are the i-th singular vector and value of matrix A (for $i = 1, \ldots, n$).

Note. The Singular Value Decomposition is not unique.

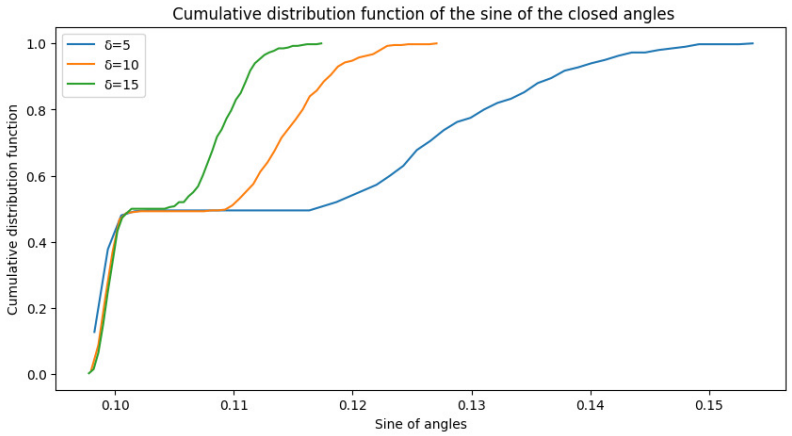


Figure 1.2.1: Sine of closed angles of the first singular vector of A and $A + E$ with a simple two-rank matrix A and Bernoulli matrix E.

We will now examine how the first singular vector of the signal matrix behaves differently from the new low-rank matrix, when the noise matrix follows a Bernoulli distribution, i.e.

$$P(E_{ij} = 1) = P(E_{ij} = -1) = \frac{1}{2}$$

with independent entries. In creating Figure 2.1.1, we simulated 400 independent Bernoulli matrices and added three simple deterministic sparse matrices to them, each containing only four nonzero components. These matrices had dimensions of 400×400, with a rank of two. We computed the first singular vectors of both the deterministic matrices and the new matrices (the

deterministic matrices plus the Bernoulli matrices). Next, we calculated the sine of the closed angles between these vectors and plotted their cumulative distribution functions. Figure 2.1.1 resembles the figure from the article by Wang, Vu, and O'Rourke in [17] on page 31. This is because Theorem 9 in the article states that if the difference between the first and second singular values of the deterministic matrix is sufficiently large, then the sine of the closed angles between these vectors will be small with high probability. However, a difference is noticeable between our figure and the diagram in the article: the cumulative distribution function converges to 1 more quickly in my figure. We believe this is because we used different deterministic matrices; our matrices were simple sparse ones, which likely led to faster convergence in this case.

In Figure 2.1.2, another cumulative distribution function is shown. In this case we simulated a Wishart matrix with rank 2 as the data matrix. Then we added 100 independent random Bernoulli matrices to it, resulting in 100 new matrices. Finally, we calculated the sine of the closed angles between the first singular vector of my Wishart matrix and the first singular vectors of the new matrices. The cumulative distribution function represents these sine values. Here, one can observe that the convergence to 1 is somewhat slower compared to the previous case. This could be because the Wishart matrix is more complex. It can be verified that the cumulative distribution function reaches 1 at approximately 0.16.

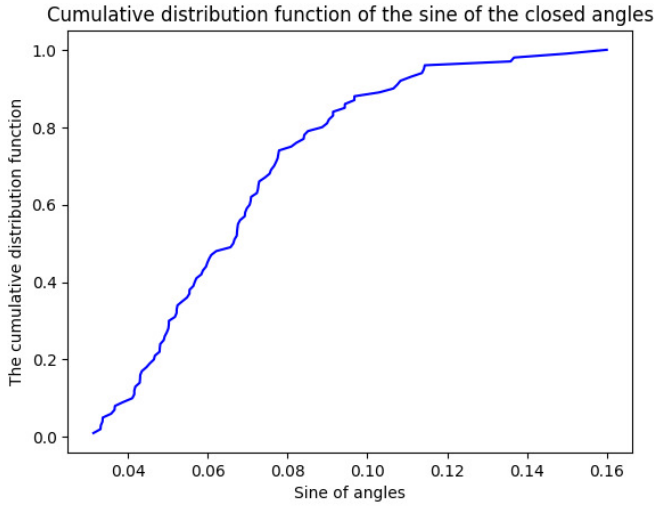


Figure 1.2.2: Sine of closed angles with Wishart matrix A and Bernoulli matrix E.

In Figure 2.1.3, we again simulated 3 deterministic matrices and added 400 independent random Bernoulli matrices to them, similarly to the first case. The difference here is that we worked with two rank matrices that are less sparse. The first two columns were linearly independent, and the remaining columns were linear combinations of these two. It can be observed that these matrices are now more similar to one another, so the difference between the

first and second singular values does not play as important a role as it did in the first case.

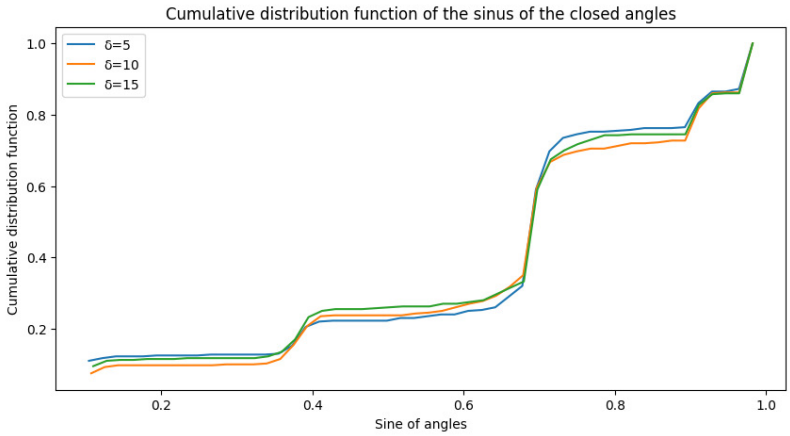


Figure 1.2.3: Sine of closed angles with more complex two-rank matrix A and Bernoulli matrix E.

1.2.1 l^∞ eigenvector bounds

To study the strength of a theorem on perturbed low-rank matrices (as in [10]), we conducted simulations for the following problem. In this setup, we assume that we have $A' = A + E$, where A is a signal matrix and E is the sum of a sparse and a noise matrix. The matrix E was always a Bernoulli matrix, and the signal matrix A was drawn from either a normal or Wishart distribution.

We performed singular value decomposition (SVD) for both A' and A, obtaining the matrices V' and V, where

$$A' := U'\Sigma'V'^{T} = \sum_{i=1}^{n} \sigma_i' u_i' v_i'^{T} \qquad \text{and} \qquad A := U\Sigma V^{T} = \sum_{i=1}^{n} \sigma_i u_i v_i^{T}$$

where

$$U, U' \in \mathbb{R}^{d_1 \times n} \qquad \text{and} \qquad V, V' \in \mathbb{R}^{d_2 \times n}.$$

Let $\sigma_1, \ldots, \sigma_n$ be the singular values of A. We define $\mu(U)$ and $\mu(V)$ as follows:

$$\mu(U) := \frac{d_1}{n} \cdot \max_i \sum_{j=1}^{n} U_{ij}^2 \qquad \text{and} \qquad \mu(V) := \frac{d_2}{n} \cdot \max_i \sum_{j=1}^{n} V_{ij}^2.$$

Next, we define A_r, the best rank-r approximation of A under the Frobenius norm:

$$A_r := \sum_{i=1}^{r} \sigma_i u_i v_i^{T}.$$

For completeness, we define two norms for a matrix $M = [M_{ij}] \in \mathbb{R}^{d_1 \times d_2}$ as follows:

$$\|M\|_{\max} := \max_{ij} |M_{ij}|, \qquad \|M\|_{\infty} := \max_{i} \sum_{j=1}^{d_2} |M_{ij}|.$$

Now we can formulate our

Theorem 1.8 (cf. [10]) We suppose that $\delta > \|E\|$ and $\sigma_r - \varepsilon = \Omega(r^3\mu^2\|E\|_\infty)$, where $\varepsilon := \|A - A_r\|_\infty$. If A is symmetric and for any $i = 1, \dots r$ the interval $[\sigma_i - \delta, \sigma_i + \delta]$ does not contain any singular values of A other than σ_i, then

$$\|V' - V\|_{\max} = \mathcal{O}\left(\frac{r^4\mu^2\|E\|_\infty}{(\sigma_r - \varepsilon)\sqrt{n}} + \frac{\sqrt{r^3\mu}\|E\|_2}{\delta\sqrt{n}}\right).$$

We note that $f = \Omega(g)$ means that $f = \mathcal{O}(g)$ and $g = \mathcal{O}(f)$.

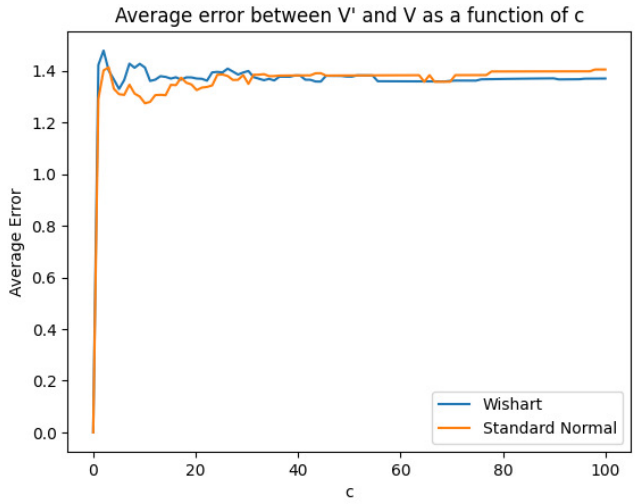


Figure 1.2.4: Simulation results for Theorem 1 with $0 \leq c \leq 100$.

To illustrate Theorem 1.8, we created two diagrams to examine the accuracy of the upper bound provided by the theorem. I generated E from a Bernoulli distribution and A from both a Wishart and a standard normal distribution, repeating this process twenty times independently. For each $c \in [0,100]$, we computed the singular value decomposition of $(A' = A + c \cdot E, A)$, obtaining the matrices (V', V). Then we plotted the mean of $\|V' - V\|_{\max}$ over the twenty cases as a function of c. In Figure 2.1.4, one can observe that the choice of distribution does not significantly impact the results, as the norms of the differences exhibit similar behavior for both distributions.

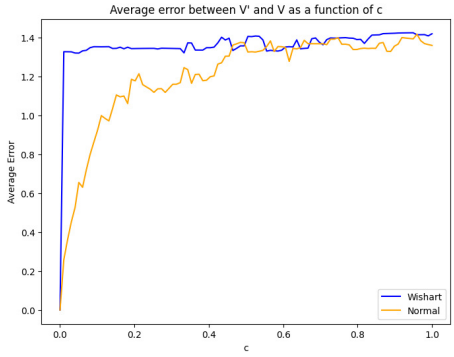


Figure 1.2.5: Simulation results for Theorem with $0 \leq c \leq 1$.

Expanding our simulation to the interval $[0,1]$, we can observe in Figure 2.3.1 that the choice of distribution does have an impact. The standard normal distribution reaches an average error of 1.3 between V' and V at $c = 0.5$ significantly more slowly than the Wishart distribution. Naturally, both curves start from zero, since for $c = 0$, we have $V' = V$ due to the uniqueness of the singular value decomposition. It is also evident that the condition

$$\sigma_r - \varepsilon = \Omega(r^3\mu^2\|E\|_\infty)$$

does not hold for the Wishart distribution. Specifically, when c is close to zero, the average error between V' and V does not exhibit linear growth. Since $\|cE\|_\infty = c \cdot \|E\|_\infty$, the upper bound should depend only linearly on the quantities in the Big-O notation when computing the average error using $c \cdot E$ instead of E.

Chapter 2

Stochastic Block Model Reconstruction with Random Error

The *Stochastic Block Model* (SBM) is a widely used probabilistic model for understanding community structures in networks. It assumes that nodes belong to hidden groups and that the probability of connections depends on group membership. By modeling these interactions, SBM reveals meaningful patterns in network data.

A given network is represented as a graph $G = (V,E)$ with n nodes, where each node i is assigned to a latent community z_i. The community memberships z_i are typically unknown and must be inferred from observed connections, making SBM a powerful tool for network analysis.

The probability of an edge between two nodes is governed by a block matrix P, where P_{kl} represents the probability of a link between nodes in groups k and l. If within-group probabilities are high compared to between-group probabilities, the network exhibits strong community structure, which SBM effectively captures.

A key advantage of SBM is its ability to model different types of networks. It can describe assortative networks, where nodes are more likely to connect within their own groups (cf. [13]), or disassortative networks, where inter-group links dominate (cf. [16]). It can also handle hierarchical structures, making it applicable to real-world networks.

Various extensions of SBM exist to better capture network complexities. The *degree-corrected SBM* (DC-SBM) accounts for degree heterogeneity, allowing nodes with different connectivity levels to be modeled more accurately (cf. [12]). The *hierarchical SBM* (hSBM) represents multi-level structures, where communities exist within larger communities, capturing nested relationships (cf. [4]).

Inference methods for SBM include Bayesian approaches, expectation-maximization algorithms and spectral clustering techniques. These methods enable researchers to identify latent communities, detect anomalies and predict missing links in large-scale networks. (cf. [8]).

SBM has applications in diverse fields, including social network analysis, biological systems and financial networks. By leveraging probabilistic modeling, it provides a rigorous framework for studying the modular organization of networks. Its flexibility and extensibility make it a fundamental tool in network science (cf. [3]).

In the SBM, edges are typically sampled independently from one another, and in our simulation, we assumed that the community consists of two latent groups.
After the introduction, we can provide a precise mathematical definition of the SBM.

Definition 2.1 Let $G = (V,E)$ be an undirected graph with $|V| = n$ vertices. The SBM generates a random graph in two steps:
(1) **Community assignment:** Each node $i \in \{1,\dots,n\}$ is independently assigned to one of r communities:

$$z_i \sim \text{Categorical}(\pi), \quad \text{where } \pi = (\pi_1,\dots,\pi_r), \quad \sum_{k=1}^{r} \pi_k = 1.$$

(2) **Edge generation:** For each pair $i < j$, the edge exists with probability

$$\mathbb{P}[(i,j) \in E] = B_{z_i,z_j}, \quad \text{where } B \in [0,1]^{r\times r} \text{ is a symmetric matrix.}$$

2.1 Simulations on the randomness of classification (histograms) and the misclassification rate

We would now like to study in more details, within four subsections, the results obtained by Abbe, Fan, Wan, and Zhong regarding the Stochastic Block Model in their 2020 paper (cf. [1]). Throughout the paper, it is assumed that the relevant community consists of two distinguishable groups. Let x be the separating vector representing the two groups. More precisely, we assume that $x \in \{1,-1\}^n$, where the i-th coordinate of x is 1 if the i-th vertex belongs to the first group (I), and -1 if it belongs to the second group ($J = V \setminus I$). We aim to estimate this vector by $\hat{x} \in \{1,-1\}^n$, with as small an error as possible. The algorithm described in the paper is quite simple, it consists of computing the second eigenvector of the random adjacency matrix A:

- Compute u_2, the eigenvector of A corresponding to its second largest eigenvalue λ_2.
- Set $\hat{x}^i := \text{sgn}(u_2^i)$.

The article specifically emphasizes that the entries of the second eigenvector also reflect the quality of the separation. When the clusters are well separated, these entries can be distinctly grouped into two sets. The corresponding theorem guarantees that this yields a good estimate:

Theorem 2.2 (cf. [1]) We assume that the distribution of the entries of the adjacency matrix of our random graph on n vertices looks like this (with $a,b > 0$ constants, and $0 < q < p < 1$):

$$P(A_{ij} = 1) = \begin{cases} p, & \text{if } i \in I \text{ and } j \in I, \text{ or } i \in J \text{ and } j \in J, \\ q, & \text{otherwise} \end{cases}$$

where

$$p := a \cdot \frac{\ln(n)}{n} \qquad \text{and} \qquad q := b \cdot \frac{\ln(n)}{n}.$$

If $\sqrt{a} - \sqrt{b} > \sqrt{2}$ then there exist an $\eta(a,b) > 0$ and $s \in \{1, -1\}$ such that with probability $1 - o(1)$

$$\sqrt{n} \cdot \min_{i \in [n]} \left(s \cdot x_i \cdot u_2^i \right) \geq \eta(a,b)$$

holds. And if $0 < \sqrt{a} - \sqrt{b} \leq \sqrt{2}$, the misclassification rate will not be too high on average:

$$\mathbb{E}\left[\min_{s \in \{\pm 1\}} \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{x_i \neq s\hat{x}_i\}} \right] \leq n^{-(1+o(1))\frac{(a-b)^2}{2}}.$$

Theorem 2.1 implies that with high probability, the coordinates of the second eigenvectors will have the same sign as the coordinates of the separating vector x, so $\operatorname{sgn}(u_2)$ will be close to x. We need to take the minimum in $s \in \{1, -1\}$, because the opposite of an eigenvector is also an eigenvector, and due to symmetry, it does not matter whether we identify the first group with $+1$ or with -1.

The natural definition of the misclassification rate when estimating x with $\hat{x}$:

$$r(x,\hat{x}) := \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}_{\{x_i \neq \hat{x}_i\}}$$

We tested the algorithm written in the article in a specific case with a fixed number of vertices. We took $n = 600$ vertices, with the first 300 vertices belonging to the first group and the rest to the second. Edges appeared with a probability of 0.55 between two vertices in the same group and with a probability of 0.43 between vertices in different groups. We plotted a histogram to show how the coordinates of $\sqrt{n} \cdot u_2$ behave.

In the second case, edges appeared with a probability of 0.65 between two vertices in the same group, and with a probability of 0.43 between vertices in different groups. It is evident that now the coordinates of $\sqrt{n} \cdot u_2$ are more separated from zero, so the sign of u_2 depends less on randomness, resulting in more confident decisions when grouping the vertices. This corresponds to the expectation that, since $p - q$ is larger than in the previous case, it becomes easier to reconstruct the groups from the random edges, as we can see it in Figure 2.1.1 and in Figure 2.1.2.

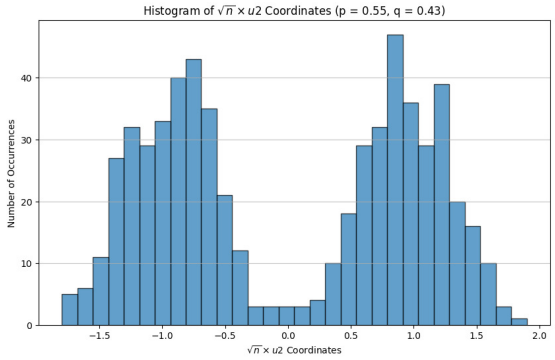


Figure 2.1.1: Histogram of $\sqrt{n} \cdot u_2$ coordinates $(p = 0.55, q = 0.43)$

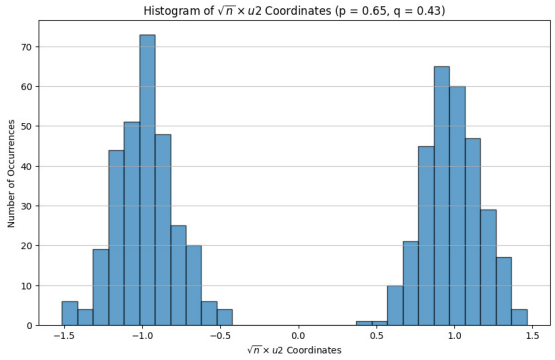


Figure 2.1.2: Histogram of $\sqrt{n} \cdot u_2$ coordinates $(p = 0.65, q = 0.43)$

We calculated the average misclassification rates in 8 different cases by generating graphs independently ten times with the appropriate edge probabilities. The rows of the table corresponded to the edge probabilities between vertices in the same group (p_{in} values), while the columns corresponded to those in different groups (p_{out} values). As expected, the closer these two numbers are, the more difficult it was to reconstruct the groups from the random edges, leading to an increased misclassification rate. If the difference between the two numbers is at least 0.06, we still classified 85% of the vertices correctly.

After that we did not change the 8 cases, nor did we alter the graphs. We ran the algorithm on the true separating vector (where the first 300 coordinates are 1 and the remaining 300 coordinates are -1) and calculated the estimated separating vector of the algorithm, recording the average of their L^2-distances for the 10 graphs in the 8 cases. The L^2-distance between the separating and estimated vector can be estimated by averaging such observations:

$$\|x - \hat{x}\| = \sqrt{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}.$$

The article mentions that the norm of the difference between the two vectors does not necessarily measure the quality of classification well. This can also be observed from the table; several times,

when the p_{in} values and the p_{out} values became closer to each other, in principle, it would have been more difficult to identify the groups from the graphs, yet the distance between the vectors in L^2-norm was still smaller even with averaging, despite the fact that the misclassification rates increased on average.

Figure 2.1.3: The average of the misclassification rates

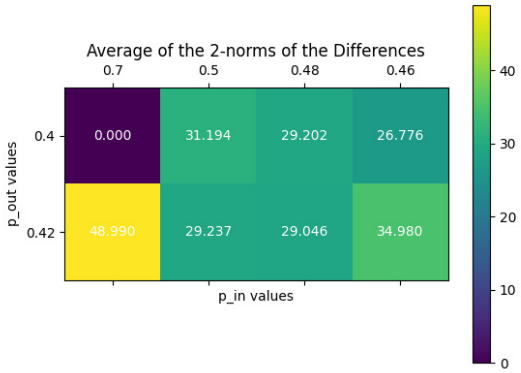


Figure 2.1.4: The average of the 2-norms of the differences

Figure 2.1.3 and Figure 2.1.4 illustrate the average misclassification probability and the average distance (in the Euclidean norm) between the estimated separating vector and the true separating vector across ten randomly and independently sampled graphs, given appropriate edge probabilities. For these histograms and misclassification rates, we obtained surprisingly good results, despite using only small sample sizes. Since our graphs had 600 vertices, even a sample size of ten graphs significantly increased the runtime.

2.2 $\mathbb{Z}_2$ synchronization

As noted in the article by Abbe, Fan, Wan and Zhong, the $\mathbb{Z}_2$ synchronization problem is closely related to the Stochastic Block Model task. In the latter, the goal is to cluster the vertices of a graph into groups based on random edge connections. In the $\mathbb{Z}_2$ synchronization problem, we observe noisy versions of random ± 1 values and aim to recover the original signal by filtering out the noise, which is typically drawn from a normal distribution.

Definition 2.3 We assume that we know the random matrix Y, generated as follows:

$$Y_{ij} = x_i \cdot x_j + \sigma \cdot W_{ij}, \quad \text{where}$$

$$x \in \{\pm 1\}^n, \quad i < j \Rightarrow W_{ij} \sim N(0,1), \quad \text{and}$$

$$\sigma > 0, \quad W_{ii} = 0 \quad W_{ij} = W_{ji}.$$

Let us further assume that variables $\{W_{ij} : i < j\}$ are independent from one another. Our aim is to recover x from Y.

Our algorithm that solves the problem is very similar to the algorithm of the Stochastic Block Model; however, here we need to work with the first (not the second!) eigenvector of Y:

1. Compute the leading eigenvector of Y, denoted by u;
2. Take the estimate $\hat{x}_i := \operatorname{sgn}(u_i)$.

Our next theorem states that the threshold for exact recovery is $\sigma = \sqrt{\dfrac{n}{2\log n}}$, and exact recovery is achievable for noise levels smaller than this.

Theorem 2.4 (cf. [1]) We suppose that for some $\varepsilon > 0$

$$\sigma \le \sqrt{\frac{n}{(2+\varepsilon)\log n}}$$

holds. Then, with probability $1 - o(1)$, the leading eigenvector u of Y with unit ℓ_2 norm satisfies

$$\sqrt{n} \cdot \min_{i \in [n]} \{s \cdot x_i \cdot u_i\} \ge 1 - \sqrt{\frac{2}{2+\varepsilon}} + \frac{C}{\sqrt{\log n}},$$

for a suitable $s \in \{\pm 1\}$, where $C > 0$ is an absolute constant.

According to Theorem 2.3, with high probability, the coordinates of x and u will all have the same sign for nonzero elements, so $\hat{x}$ equals to x. The factor $s \in \{\pm 1\}$ is included in the theorem because, due to symmetry, it does not matter whether we identify the first group with $+1$ or -1.

2.3 Stochastic Block Model combined with noise

	Sigma	M.R.
0	0.0	0.008
1	0.2	0.017
2	0.4	0.045
3	0.6	0.102
4	0.8	0.155
5	1.0	0.275

Figure 2.3.1: The average misclassification rates with different noises, $p_{in} = 0.5$, $p_{out} = 0.4$

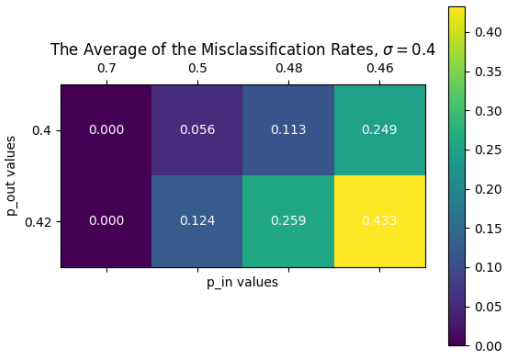


Figure 2.3.2: Error rates with noise (sigma = 0.4) – second eigenvector

Figure 2.3.1 illustrates the following process: we fixed two edge probabilities, one for within-group connections and another for between-group connections, and then independently generated ten random graphs, each with 600 vertices, using these probabilities. To the adjacency matrix of each graph, we added a scaled version of a 600×600 symmetric matrix sampled from a standard normal multivariate distribution, with the scaling factor σ varying according to the noise levels shown on the left-hand side of the table. This differs from our $\mathbb{Z}_2$ synchronization theorem because, in that case, we added noise to the matrix $x^\top x$ aiming to recover x from the noisy matrix. Here, the error could arise from two sources: first, the edges themselves are random; second, we could only observe the adjacency matrix in a noisy environment. Thus, we applied the Stochastic Block Model algorithm to the noisy adjacency matrix and evaluated the accuracy of the reconstruction with the added noise. For our case, $\sigma = 0.4$ was the highest noise level where the misclassification rate did not exceed 5%. For larger noise levels, the results deteriorated significantly. We also plotted the misclassification rate for different edge probabilities with a

sigma of 0.4. Due to the noise, this turned out slightly worse, which can be seen in Figure 2.3.2. When we implemented it ourselves to compute the misclassification rate in the case where the simulation uses noise of the same type as in the theorem, we obtained a misclassification rate of 45%. This probably occurred because, in our simulation, we grouped the vertices based on the sign of the second eigenvector of the noisy adjacency matrix, rather than on the sign of the leading eigenvector of the noisy matrix $x^\top x$. (The i-th row and j-th element of $x^\top x$ is 1 if the i-th vertex and the j-th vertex belong to the same group, and -1 if they belong to different groups, x is the separating vector.)

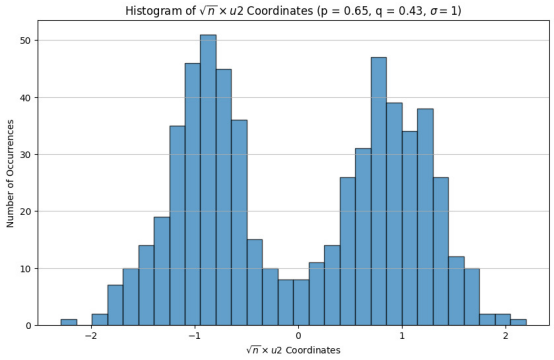


Figure 2.3.3: The histogram of the coordinates of $\sqrt{n} \cdot u_2$ with $p_{in} = 0.65$, $p_{out} = 0.43$, $\sigma = 1$

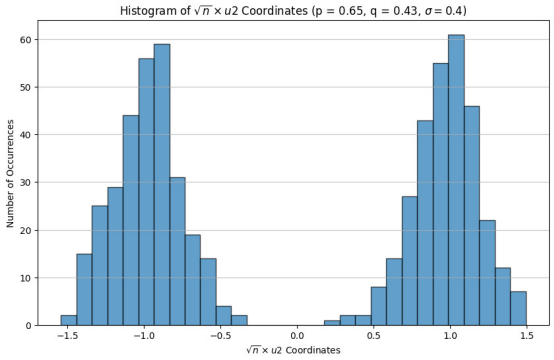


Figure 2.3.4: The histogram of the coordinates of $\sqrt{n} \cdot u_2$ with $p_{in} = 0.65$, $p_{out} = 0.43$, $\sigma = 0.4$

In Figure 2.3.3 and Figure 2.3.4 we considered the same problem, calculated the second eigenvector of the noisy version of the adjacency matrix, and assigned the vertices to groups based on their sign. It can be observed that with larger noise ($\sigma = 1$), the eigenvector coordinates are less separated from 0, making the classification of a vertex into the correct group more dependent on randomness compared to when smaller noise is chosen ($\sigma = 0.4$).

2.4 Application on real data

We tested the Stochastic Block Model algorithm on a deterministic graph as well, where the edge between two vertices does not depend on randomness. We downloaded the graph from [22]. We ran the algorithm, which works well on random graphs, on the graph and divided the vertices into two groups. The graph contained 1,226 vertices and 2,615 edges. An interesting question is how much stronger certain properties of graphs are within the groups compared to the entire graph. To address this question, we can define the concepts of edge density:

Definition 2.5 The density of a graph with n vertices and m edges is $\frac{m}{\binom{n}{2}}$. This indicates how dense the edges are in the graph relative to the complete graph.

and clustering coefficient:

Definition 2.6 For a graph $(G = (V,E))$, the clustering coefficient of a vertex $v \in V$ is defined as follows:

$$C(v) = \frac{|\{\{u,w\} \in E : u,w \in N(v)\}|}{\binom{\deg(v)}{2}},$$

where $N(v)$ is the set of neighbors of vertex v, and $\deg(v)$ is the degree of vertex v. The overall clustering coefficient of the graph is the average of the clustering coefficients of all vertices:

$$C = \frac{1}{n} \cdot \sum_{v \in V} C(v).$$

This concept also describes the cohesion of the elements of a graph. Suppose the vertices of the graph represent people, and there is an edge between two vertices if the corresponding people know each other. The clustering coefficient in the graph will be high if many of a given person's acquaintances know each other as well. These concepts can similarly be defined for a subset of the vertices of a graph.

After the definitions, we can illustrate the graph on which we ran the Stochastic Block Model. The graph and the two groups generated by the model are depicted in 2.5.1. We expect higher edge density and clustering coefficient within the groups.

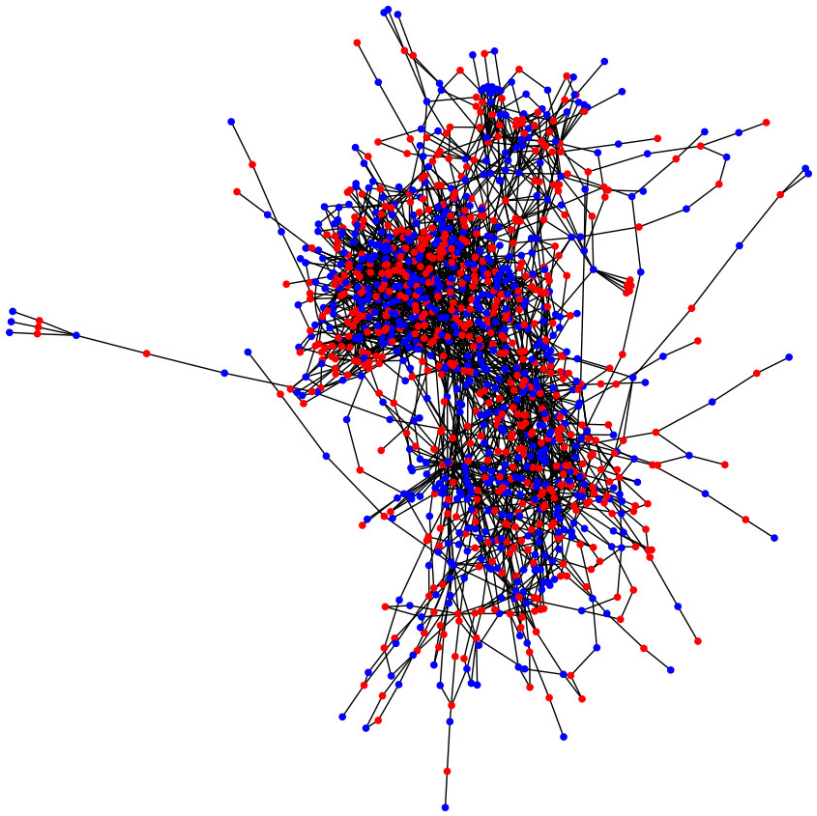


Figure 2.4.1: The two groups of our graph (adjacency matrix)

The description of our graph is as follows: "This network was constructed from the USA's FAA (Federal Aviation Administration) National Flight Data Center (NFDC), Preferred Routes Database. Nodes in this network represent airports or service centers and links are created from strings of preferred routes recommended by the NFDC." It is evident that the natural expectations related to the algorithm are met, namely, the edge density and clustering coefficient are higher within the groups than in the entire graph:

Number of nodes in group 1: 616
Number of nodes in group 2: 610
Edge density in the entire graph: 0.0032093751040383526
Clustering coefficient in the entire graph: 0.06750771494491796
Edge density in Group 1: 0.0054465209587160807

Edge density in Group 2: 0.006389001049826375
Clustering coefficient in Group 1: 0.09009508348794063
Clustering coefficient in Group 2: 0.10314207650273224

As we expected, the Stochastic Block Model algorithm doesn't perform too badly, as both the edge density and the clustering coefficient are higher when we narrow the graph down to the groups. In other words, among the American cities that belong to the same group, there is a higher probability of flights between them. Moreover, if a city has flights to two other cities, there is a higher likelihood of a flight between those two cities, provided these cities are in the same group. In the second group, the cohesion is slightly stronger than in the first one. However, the algorithm is not perfect. The histogram of the coordinates of the second eigenvector of the adjacency matrix unfortunately doesn't separate well enough from zero, as shown in Figure 2.4.2. This means that randomness plays a significant role in whether the two groups are sufficiently separated from each other.

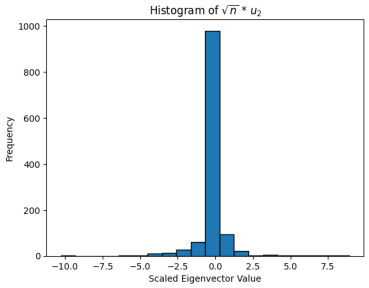


Figure 2.4.2: The coordinates of the second eigenvector, real data

2.5 Stochastic Block Model for sparse graphs

In the following, we will study the article by Ludovic Stephan and Laurent Massoulié published in 2018 (see [21]), and examine its main results through simulations. They also worked with the Stochastic Block Model, but in their paper, they generally assumed the existence of more than two groups, and their results perform well on sparser graphs. Although their method for two groups resembles the algorithm described by Abbe, Fan, Wan, and Zhong ([1]) in their 2020 paper, they worked with different matrices and their eigenvalues and eigenvectors. To understand their algorithm, we need to define certain matrices, whose leading eigenvectors will play a key role.

Definition 2.7 Let G be any graph, and let ℓ be a positive integer. We define two matrices associated with G:

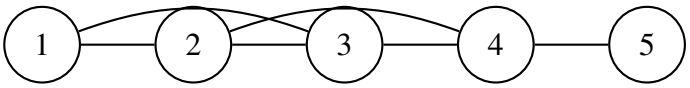
(i) the *path expansion matrix* $B^{(\ell)}$ (studied in Massoulié (2014)), whose (i,j) coefficient counts the number of self-avoiding paths (that is, paths that do not go through the same vertex twice) of length ℓ between i and j,

(ii) the *distance matrix* $D^{(\ell)}$, defined by

$$D_{ij}^{(\ell)} := \begin{cases} 1, & \text{if } d(i,j) = \ell, \\ 0, & \text{otherwise} \end{cases}$$

where d denotes the usual graph distance.

Example. We provide the two matrices above for a specific graph when considering paths of length 3.



The path expansion matrix $B^{(3)}$ counts the number of self-avoiding paths of length 3 between each pair of nodes, while the distance matrix $D^{(3)}$ indicates whether the graph distance between two nodes is exactly 3:

$$B^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 2 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{bmatrix}, \qquad D^{(3)} = \begin{bmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

In this paper, the authors considered multiple groups and studied, with the help of the eigenvectors of $B^{(\ell)}$ and $D^{(\ell)}$, to what extent it is possible to construct an algorithm that performs better than random guessing; in other words, assigning each vertex to a group based on prior information.

Let the prior distribution be denoted by $\boldsymbol{\pi}$. If there are $\mathbf{r}$ groups, then under π, each vertex is assigned to group k with probability $\pi(k)$, for $k = 1,2,\ldots,r$. (Under the uniform distribution, this corresponds to random guessing.)

In our setting, we will denote the true separating vector by $\boldsymbol{\sigma}$, which corresponds to the vector denoted by x in the previous two-group analysis.

Let the entry in the i^{th} row and j^{th} column of the matrix $\mathbf{W}$ represent the probability that an edge exists between two vertices, where the first vertex belongs to group i, and the second one belongs to group j (where $1 \leq i,j \leq r$). This matrix has its maximum values along the diagonal, since edges are more likely to occur between vertices within the same group; in other words, the connections are stronger within groups. After introducing the key concepts, we can now explore the various aspects from which the article defines the Stochastic Block Model.

Definition 2.8 Let $r \in \mathbb{N}$ be fixed, let W be an $r \times r$ symmetric matrix with nonnegative entries, and let π be a probability vector on $[r]$. A random graph $G = (V,E)$ with $|V| = n$ is said to be distributed according to the *Stochastic Block Model* (SBM) with r blocks and parameters (W, π) if:

(i) each vertex $v \in V$ is assigned to a type $\sigma(v)$ sampled independently from π,

(ii) any two vertices $u,v \in V$ are joined with an edge randomly and independently from every other edge, with probability

$$\min\left\{\frac{W_{\sigma(u),\sigma(v)}}{n}, 1\right\}.$$

The probability of an edge between two vertices is given by the corresponding entry of W scaled by $1/n$, so that the expected degree of each vertex remains asymptotically constant as n approaches infinity. The article does not deal with exact recovery, as it focuses solely on sparse graphs. In such cases, identifying the underlying communities from random edges is more challenging, since the graph contains fewer dense subgraphs. As a result, we can only hope for a weaker notion of reconstruction, which is referred to as *partial reconstruction*.

Partial reconstruction is defined as the case when the *liminf* of the correct classification rate is strictly greater than the proportion of nodes that would be correctly classified under random guessing, that is, according to the relative size of the groups. This means that, for sufficiently large n, the algorithm assigns nodes to their correct groups with higher accuracy than random assignment based solely on group proportions:

Definition 2.9 Let σ be the true type assignment, and $\hat{\sigma}$ an estimate of σ. The *empirical overlap* between σ and $\hat{\sigma}$ is defined as

$$\mathrm{ov}(\sigma,\hat{\sigma}) = \max_{\tau\in S_r}\left(\frac{1}{n}\sum_{v=1}^{n}\mathbf{1}_{\{\hat{\sigma}(v)=\tau(\sigma(v))\}}\right) - \max_{k\in[r]}\pi_k,$$

where S_r is the set of permutations of $[r]$.

For a given algorithm that produces estimates $\hat{\sigma}$ for all n vertices, we say that the algorithm achieves *partial reconstruction* if

$$\liminf_{n\to\infty}\mathrm{ov}(\sigma,\hat{\sigma}) > 0 \quad \text{with high probability.}$$

The main result of the article is that the groups can be partially recovered from the random edges using the eigenvectors of $B^{(\ell)}$ and $D^{(\ell)}$. Although the article does not provide an explicit algorithm, it becomes clear that if certain coordinates of the relevant eigenvector (the second eigenvector of either $B^{(\ell)}$ or $D^{(\ell)}$) behave similarly, then the corresponding vertices are likely to belong to the same group. In the case of two groups, such similar behavior can be, for example, having the same sign. In that case, we essentially recover the algorithm of Abbe, Fan, Wan and Zhong. In their work, the sign of the second eigenvector of the random adjacency matrix determined the classification; here, it is the sign of the second eigenvector of $B^{(\ell)}$or $D^{(\ell)}$ that plays the same role.

Theorem 2.10 (cf. [21]) Assume that $\pi \equiv 1/r$ and that W is a stochastic, positive regular matrix. Let μ_1 and μ_2 denote the two largest eigenvalues of W, and suppose that the following condition holds:

$$\mu_2^2 > \mu_1.$$

Then there exists an algorithm, based solely on an eigenvector of $B^{(\ell)}$ associated with its second largest eigenvalue, that achieves partial reconstruction whenever $\ell \sim \delta\log(n)$ for small enough δ.
The same algorithm also achieves partial reconstruction when applied to $D^{(\ell)}$ instead of $B^{(\ell)}$, under the same condition on ℓ.

In the above theorem, the second eigenvector of $B^{(\ell)}$ is also considered. In the case of two groups, the sign of this eigenvector determines which group each vertex belongs to. We now compare this algorithm to the one where the sign of the eigenvector of the adjacency matrix was used for classification. The comparison is performed on the same deterministic graph as in the previous subsection. The sign of the second eigenvector of the path extension matrix divided the vertices into two groups as follows:

Number of nodes in group 1: 718
Number of nodes in group 2: 508
Edge density in the entire graph: 0.0032093751040383526
Clustering coefficient in the entire graph: 0.06750771494491796
Edge density in Group 1: 0.005077640897736235
Edge density in Group 2: 0.006367547251859789
Clustering coefficient in Group 1: 0.09712993577158519
Clustering coefficient in Group 2: 0.10473646477136406

We found that the two algorithms classify the elements into clusters with similar strength; however, interestingly, the assignments were entirely different. Specifically, the program computed that only 53.67% of the vertices were assigned to the same group by both procedures. By the way, during the simulation, we divided the vertices into two parts by examining paths of length $\mathfrak{l} = 3$. That is, if there are many paths of length 3 between two vertices, then they are more likely to be assigned to the same group.

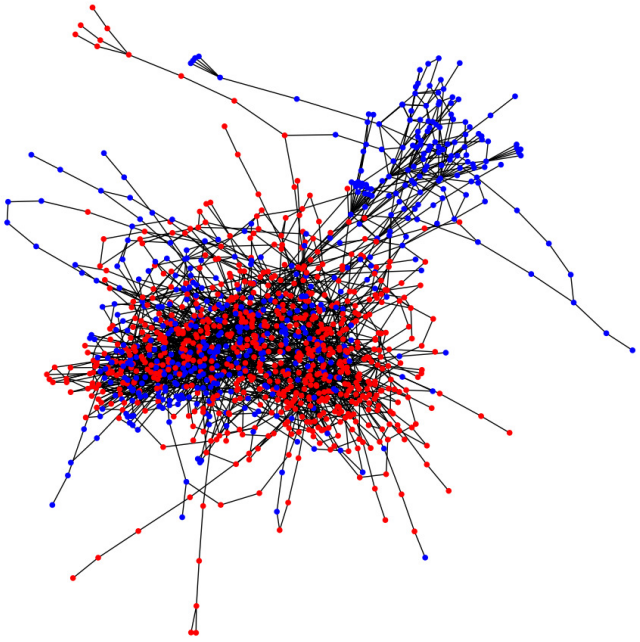


Figure 2.5.1: The two groups of our graph (path expansion matrix)

2.5.1 Simulation of Edge Perturbation and Other Results

We now proceed to discuss the theorem that serves as the basis for our simulations. In the following problem, we are faced with a form of perturbation, the resolution of which presents

several difficulties. Since even for large n, each vertex has on average only a constant number of neighbors, the graph remains sparse. As a result, recovering the underlying communities becomes more difficult, and according to the previous theorem, we can only guarantee partial reconstruction. In this setting, perturbation refers to an adversary that is allowed to either delete edges incident to certain vertices or insert edges between them. Our task is to perform partial reconstruction even after such edge modifications have occurred in a sparse graph. We now formally define this problem.

Definition 2.11 Let $\gamma := \gamma(n)$ be a positive integer, and let G be any graph on n vertices. An adversary of strength γ is allowed to arbitrarily add and remove edges, as long as the number of affected vertices (i.e., vertices that are endpoints of altered edges) is at most γ.

The following theorem, which we validated through simulation, states that if the number of vertices whose incident edges are modified remains sufficiently small compared to the total number of vertices, then partial recovery can still be achieved.

Theorem 2.12 (cf. [21]) Under the same assumptions as Theorem 2.9, let G be a graph generated via the Stochastic Block Model, and let $\widetilde{G}_\gamma$ be the graph obtained after perturbation by an adversary of strength γ.
Then, assuming

$$\gamma = o\left(\frac{\left(\frac{\mu_1}{\mu_2}\right)^{\ell/2}}{\log(n)}\right)$$

the algorithm of Theorem 2.9 still achieves partial reconstruction on $\widetilde{G}_\gamma$.
The above result on γ is optimal up to a factor of $\log(n)$.

The simulations were conducted as follows: the number of vertices was fixed, with $n = 1000, 2000, \ldots, 6000$, which correspond to the columns in the table. The rows correspond to different values of γ, indicating how many vertices had their incident edges modified. In the first row, a large number of vertices were perturbed, but the number was kept constant ($c = 200$). In the second row, the same constant was divided by $\log(n)$. In the third row, approximately as many vertices were perturbed as described in Theorem 2.11. Finally, the last row shows the results corresponding to the setting with zero edge modifications. In each case, we independently generated 20 random graphs, where the edges were randomly sampled and subsequently perturbed. Then we calculated the average misclassification rates for the respective scenarios. As in the previous section, we assumed here as well that the community has a latent two-group structure.

$\gamma \backslash n$	$n = 1000$	$n = 2000$	$n = 3000$	$n = 4000$	$n = 5000$	$n = 6000$
$\gamma = c$	0.500	0.454	0.469	0.500	0.499	0.500
$\gamma = c/\log(n)$	0.439	0.316	0.406	0.339	0.333	0.315
$\gamma = \mu_2^2/(\mu_1 \log(n))$	0.317	0.374	0.343	0.302	0.409	0.299
$\gamma = 0$	0.247	0.238	0.252	0.246	0.258	0.281

The average misclassification rate for different values of γ (number of modified vertices) and different values of n (number of vertices).

As shown in the first table, the fewer vertices have their incident edges modified, the lower the average misclassification rate becomes. Since both p_{in} and p_{out} (i.e., $\frac{W}{n}$) converge to zero as the number of vertices tends to infinity, recovering the community structure becomes increasingly difficult for larger n. This explains why the corresponding theorem guarantees only partial recovery in this regime.
We computed the empirical variance of the misclassification rate in each corresponding case, based on a single run involving 20 independently generated graphs.

$\gamma \backslash n$	1000	2000	3000	4000	5000	6000
$\gamma = c$	0.100	0.092	0.152	0.096	0.130	0.091
$\gamma = c/\log(n)$	0.230	0.206	0.123	0.094	0.162	0.118
$\gamma = \mu_2^2/(\mu_1 \log(n))$	0.223	0.263	0.127	0.129	0.150	0.104
$\gamma = 0$	0.123	0.143	0.135	0.095	0.099	0.017

Empirical standard deviation of the misclassification rate based on 20 graphs.

Although the variances are not low, the data suggest that if we perturb the edges of fewer vertices, we can recover the groups more accurately. During the simulation, the computation of statistics using 20 graphs per case significantly increased the runtime.

Finally, we performed one-sample t-tests to assess whether the simulation results were indeed significantly better than those expected from random guessing, in those cases where the edges of a certain number of vertices were modified, as specified in the theorem. The null hypothesis H_0 stated that the expected value of the misclassification rate is equal to $\frac{1}{2}$. We tested this hypothesis at a significance level of $\alpha = 0.05$. To this end, we computed the corresponding p-values for each case, and whenever the p-value was less than 0.05, we were able to reject the null hypothesis. This allowed us to establish the alternative hypothesis $H_1 : mr < \frac{1}{2}$. Due to the symmetry of the problem $\left(\text{i.e., the existence of only two groups with } \pi(1) = \pi(2) = \frac{1}{2}\right)$, we expect $mr \leq \frac{1}{2}$ under random guessing, which justifies the direction of the test.

$\gamma \backslash n$	$n = 1000$	$n = 2000$	$n = 3000$	$n = 4000$	$n = 5000$	$n = 6000$
$\gamma = c$	0.175	0.085	0.077	0.064	0.085	0.087
$\gamma = c/\log(n)$	0.006*	0.040*	0.043*	0.084	0.017*	0.039*
$\gamma = \mu_2^2/(\mu_1 \log(n))$	0.016*	0.021*	0.006*	0.045*	0.039*	0.016*
$\gamma = 0$	$6.9 \cdot 10^{-8}$ *	$2.4 \cdot 10^{-9}$ *	$4.3 \cdot 10^{-10}$ *	$3.5 \cdot 10^{-5}$ *	$4.8 \cdot 10^{-12}$ *	$1.9 \cdot 10^{-6}$ *

Testing the null hypothesis $H_0 : mr = \frac{1}{2}$ *against the alternative hypothesis* $H_1 : mr < \frac{1}{2}$ *in the different cases, with the corresponding* p*-values reported.*

In this table, asterisks indicate the cases in which the null hypothesis could be rejected. It can be observed that when a large (constant) number of vertices were perturbed, the null hypothesis was never rejected. This is not surprising, since in those cases the misclassification rate was consistently close to $\frac{1}{2}$. When the number of perturbed vertices was scaled by the logarithm of the total number of vertices, the null hypothesis was rejected in 5 out of 6 cases, indicating that the algorithm outperformed random guessing in most instances. In the case of the value of γ given in the theorem, the null hypothesis was successfully rejected in all cases. This is even more evident considering the extremely low p-values obtained when no edge modifications were applied.
In the example above, we use the following parameter values:

- $\mu_1 = 10$ (expected number of within-community edges)
- $\mu_2 = 4$ (expected number of between-community edges)
- $c = 100$ (a constant controlling the number of modified vertices)
- $p_{in} = \dfrac{\mu_1 + \mu_2}{2n} = \dfrac{7}{n}$
- $p_{out} = \dfrac{\mu_1 - \mu_2}{2n} = \dfrac{3}{n}$
- $\ell = \left\lfloor \dfrac{1}{13} \log_\alpha n \right\rfloor = \left\lfloor \dfrac{1}{13} \lg n \right\rfloor$

This raises the question of how reliable the coordinates of the eigenvector $B^{(\ell)}$ are, that is, to what extent their behavior aids in the identification of the groups. We define the matrix M as follows: $M = \Pi \cdot W$, where $\Pi = \mathrm{diag}(\pi_1, \ldots, \pi_r)$. In this case, W and M are similar, and hence they share the same eigenvalues. It can be seen that the element in the ith row and jth column of matrix M corresponds to the expected number of neighbors that a vertex in group i has in group j.

We order the eigenvalues of M (or W) in decreasing order of absolute value:

$$\mu_1 \geq |\mu_2| \geq \cdots \geq |\mu_r|.$$

To ensure successful recovery of the groups, it is essential to assume that

$$M \text{ is positive regular}, \quad \alpha := \mu_1 > 1, \quad \text{and } \tau := \mu_2^2/\mu_1 > 1.$$

We will only be interested in the first few eigenvalues of M, that is, the first r_0 eigenvalues, where r_0 is defined as follows:

$$\mu_{r_0+1}^2 \leq \mu_1 < \mu_{r_0}^2.$$

Since M is symmetric, it is diagonalizable in an orthonormal basis. Therefore, we may introduce its system of eigenvectors, which forms an orthonormal basis in $\mathbb{R}^r$:

$$\phi_i^\top M = \mu_i \phi_i^\top \quad \text{for all } i \in [r].$$

We introduce the following scalars and vectors:

$$\chi_k(v) = \phi_k(\sigma(v)), \quad \varphi_k = \frac{B^{(\ell)}\chi_k}{\|B^{(\ell)}\chi_k\|} \quad \text{for all } k \in [r].$$

The eigenvalues of $B^{(\ell)}$ are also arranged in decreasing order of absolute value:

$$\lambda_1(B^{(\ell)}) \geq \left|\lambda_2(B^{(\ell)})\right| \geq \cdots \geq \left|\lambda_n(B^{(\ell)})\right|$$

With all these preparations in place, we are now ready to state our next theorem.

Theorem 2.13 (cf. [21]) Consider a graph G generated as above, and let $\ell \sim \kappa \log_\alpha(n)$, with $\kappa < \frac{1}{12}$. Then, with probability going to 1 as $n \to +\infty$:

(i) $\lambda_k(B^{(\ell)}) = \Theta(\mu_k^\ell)$ for $k \in [r_0]$,

(ii) for $k > r_0$, $\lambda_k(B^{(\ell)}) = O\left(\log(n)^c\, \alpha^{\ell/2}\right)$ for some constant $c > 0$.

Furthermore, consider μ such that $\mu^2 > \alpha$ and μ is an eigenvalue of multiplicity d of M. Let $\varphi^{(1)}, \ldots, \varphi^{(d)}$ be an orthonormal basis of eigenvectors of M associated to μ, and $\phi^{(1)}, \ldots, \phi^{(d)}$ the vectors defined as in (9). There exist orthogonal vectors $\xi^{(1)}, \ldots, \xi^{(d)}$ in $\mathbb{R}^n$ such that the following holds:

(iii) for all i, $\xi^{(i)}$ is an eigenvector of $B^{(\ell)}$ with associated eigenvalue $\Theta(\mu^\ell)$,

(iv) there exists an orthogonal matrix $Q \in \mathcal{O}(d)$ such that

$$\|\boldsymbol{\varphi} Q - \xi\| = \mathcal{O}\left(\alpha^{\ell/2}\mu^{-\ell}\right),$$

where $\boldsymbol{\varphi}$ (resp. ξ) is the $n \times d$ matrix whose columns are the $\phi^{(i)}$ (resp. the $\xi^{(i)}$).

Our theorem has important implications for the identification of communities in random sparse graphs. The matrix M typically contains the expected number of connections between each pair of groups. If the graph is generated according to this distribution, then the eigenvectors of M reflect the community structure. Therefore, the ideal group assignment can be determined based on the first few eigenvectors of M.

However, in practice, we usually only observe the random edges and the graph itself. Our theorem states that using the matrix $B^{(\ell)}$ constructed from the graph, we do not perform the grouping significantly worse, since the important (first r_0) eigenvalues of $B^{(\ell)}$ differ from those of M only by a constant factor, independent of the randomness. If the leading eigenvalues of M and $B^{(\ell)}$ are close, then their corresponding eigenvectors will also have approximately the same directions. Moreover, the less significant eigenvalues of $B^{(\ell)}$ decay at an exponential rate.

It can therefore be observed that the columns of the matrix ξ, which contains the eigenvectors of $B^{(\ell)}$, effectively cluster the vertices into groups. The final part of the theorem states that these vectors can be well approximated by the vectors φ_k ($k = 1, \ldots, r$), which can be computed directly from the graph with fewer calculations. This is because there exists an orthogonal matrix such that, when multiplied with the matrix $\boldsymbol{\varphi}$ (whose columns are the vectors φ_k, $k = 1,2,\ldots,r$), the resulting matrix becomes close to ξ. Therefore, by applying an isometric transformation to the vectors in $\boldsymbol{\varphi}$, we obtain a set of vectors whose coordinates can be effectively used to cluster the vertices.

In the proof of the theorem, elementary linear algebraic tools, such as the Gram-Schmidt orthogonalization and the triangle inequality, are employed. Furthermore, the authors utilized the fact that the length of any arbitrary vector does not change when it is multiplied on the left by an orthogonal matrix (i.e., when it is rotated or reflected). However, the main tool of the proof was the Weyl's inequality, which we discuss in the following.

Theorem 2.14 – Weyl's inequality (cf. [23])
Let A and B be symmetric (or Hermitian) on an inner product space V with dimension n, with spectrum ordered in descending order

$$\lambda_1(X) \geq \lambda_2(X) \geq \ldots \geq \lambda_n(X), \quad X \in \{A, B, A+B\}.$$

We note that these eigenvalues can be ordered, because they are real (as eigenvalues of symmetric matrices). The following inequality holds for any integers i and j (with appropriate indices):

$$\lambda_{i+j-1}(A+B) \leq \lambda_i(A) + \lambda_j(B) \leq \lambda_{i+j-n}(A+B).$$

Proof. Since the matrices are symmetric, by the min-max theorem it suffices to show that for any subspace $W \subset V$ of dimension $i+j-1$, there exists a unit vector $w \in W$ such that

$$\langle w,(A+B)w\rangle \leq \lambda_i(A) + \lambda_j(B).$$

By the min-max principle, there exists a subspace $W_A \subset V$ of codimension $i-1$ such that

$$\lambda_i(A) = \max_{\substack{x \in W_A \\ \|x\|=1}} \langle x, Ax \rangle.$$

Similarly, there exists a subspace $W_B \subset V$ of codimension $j-1$ satisfying

$$\lambda_j(B) = \max_{\substack{x \in W_B \\ \|x\|=1}} \langle x, Bx \rangle.$$

Since the intersection $W_A \cap W_B$ has a codimension at most $i+j-2$, it must have a nontrivial intersection with any subspace $W \subset V$ of dimension $i+j-1$. Therefore, we can choose a unit vector

$$w \in W \cap W_A \cap W_B.$$

For this vector, then we have

$$\langle w, (A+B)w \rangle = \langle w, Aw \rangle + \langle w, Bw \rangle \leq \lambda_i(A) + \lambda_j(B).$$

For the second part of the inequality, let us use the fact that the eigenvalues of the negative of a symmetric matrix are the negatives of the eigenvalues of the original matrix:

$$\lambda_i(-A) = -\lambda_{n-i+1}(A).$$

Since $-A$ and $-B$ are also symmetric, we can apply the first inequality to their opposites:

$$\lambda_{i+j-1}(-A-B) \leq \lambda_i(-A) + \lambda_j(-B).$$

On the left-hand side:

$$\lambda_{i+j-1}(-A-B) = -\lambda_{n-(i+j-1)+1}(A+B) = -\lambda_{n-i-j+2}(A+B).$$

On the right-hand side:

$$\lambda_i(-A) + \lambda_j(-B) = -\lambda_{n-i+1}(A) - \lambda_{n-j+1}(B).$$

Therefore,

$$-\lambda_{n-i-j+2}(A+B) \leq -\lambda_{n-i+1}(A) - \lambda_{n-j+1}(B).$$

If we multiply it by -1, we obtain the following. For all indices $i' = n-i+1$ and $j' = n-j+1$, it holds that

$$\lambda_{i'+j'-n}(A+B) = \lambda_{n-i-j+2}(A+B) \geq \lambda_{n-i+1}(A) + \lambda_{n-j+1}(B) = \lambda_{i'}(A) + \lambda_{j'}(B).$$

Chapter 3

Application of Graph Powers in the Stochastic Block Model

In many graph-theoretic and statistical problems, it is useful to consider not only direct connections between nodes, but also indirect paths of a given length. This motivates the notion of graph powering, a technique that transforms a given graph into a new one by connecting nodes that are within a fixed distance ℓ from each other. Specifically, in the ℓ-th power of a graph, two nodes are connected if there exists a path of length at most ℓ between them in the original graph.

Graph powering has gained significant attention in the context of community detection, particularly in sparse regimes where local information is limited. By amplifying the connectivity structure, graph powers help to mitigate the noise in edge observations and enhance the signal associated with latent clusters.

In the context of the Stochastic Block Model (SBM), powering the graph before applying spectral methods can lead to improved performance in both detection and reconstruction tasks. In particular, powering helps to bridge disconnected or weakly connected regions within the same community, effectively increasing the spectral gap and improving the reliability of leading eigenvectors.

Before discussing graph powering, we present a new perspective on the Stochastic Block Model.

3.1 Spectral Properties and Weak Recovery

Before presenting the simulations, we will review the main results of [2].
The Stochastic Block Model can also be examined from a different perspective. In order to proceed, we first introduce some key concepts. We may specify the probability with which a fixed vertex is assigned to a particular group, given by the vector $p = (p_1, \ldots, p_k)$. Additionally, we can define the probability that an edge is formed between two vertices (not necessarily from

different groups) as W_{X_i,X_j}, where X_i and X_j denote the group memberships of the respective vertices.

Definition 3.1
Let $n \in \mathbb{N}^+$ denote the number of vertices, and let $k \in \mathbb{N}^+$ be the number of communities. Let $p = (p_1,\ldots,p_k)$ be a probability vector on the set $[k] := \{1,\ldots,k\}$, representing the prior distribution over the k communities. Furthermore, let $W \in [0,1]^{k\times k}$ be a symmetric matrix, where each entry $W_{a,b}$ indicates the probability that a vertex from community a connects to a vertex from community b.
We say that the pair (X,G) is drawn from the Stochastic Block Model, denoted by SBM(n,p,W), if the followings hold:

- $X = (X_1,\ldots,X_n)$ is a random vector where each component X_i is independently sampled from the distribution p, that is, $X_i \sim p$ i.i.d. for all $i \in [n]$. The variable X_i represents the community label of vertex i.
- G is a random undirected graph on n vertices, where each edge between distinct vertices i and j is included independently with probability W_{X_i,X_j}.

We also define the community sets by

$$\Omega_i = \Omega_i(X) := \{v \in [n] : X_v = i\}, \quad \text{for each } i \in [k].$$

We can also define a special case of SBM.

Definition 3.2
We define a special case of the Stochastic Block Model, denoted by $\mathrm{SBM}(n,a,b)$, where the number of communities is $k = 2$. In this setting:

1. The community prior is uniform:

$$p = \left(\frac{1}{2},\frac{1}{2}\right),$$

meaning that each vertex is independently assigned to one of the two communities with equal probability.

2. The connectivity matrix $W \in \mathbb{R}^{2\times 2}$ is symmetric, and its entries are defined as:

$$W = \begin{bmatrix} \frac{a}{n} & \frac{b}{n} \\ \frac{b}{n} & \frac{a}{n} \end{bmatrix},$$

where

- $\frac{a}{n}$ is the probability that two vertices from the **same** community are connected,
- $\frac{b}{n}$ is the probability that two vertices from **different** communities are connected.

As a result, the graph G contains two equally sized (balanced) clusters, with edge probability $\frac{a}{n}$ inside the clusters and $\frac{b}{n}$ between clusters.

Now we turn to the presentation of the main results of the paper, which are related to theorems involving spectral methods based on eigenvalues and eigenvectors. To this end, we can introduce the following new concepts, which are also related to the Stochastic Block Model. We study various random graph models with planted community structures. In each case, we consider an ensemble $\mathcal{M}(n)$ that defines a probability distribution over pairs (X,G), where

- X is a random vector in $\mathbb{R}^n$ with independent and identically distributed (i.i.d.) components, representing the hidden community labels of the n vertices,
- G is a random graph on n vertices, where the presence or absence of edges depends on the labels in X.

The central task is to recover the label vector X based solely on the observation of the graph G, that is, to infer the underlying communities from the network structure.

This work focuses on the sparse regime, where the average degree of the graph remains bounded as $n \to \infty$, and on the problem of weak recovery, which is formally defined in the sections below.

Definition 3.3 – Weak recovery
In the case of k communities, an algorithm

$$\widehat{X} : 2^{\binom{[n]}{2}} \to [k]^n$$

is said to recover the communities **with accuracy** $f(n)$ under the model $\mathcal{M}(n)$ if, for $(X,G) \sim \mathcal{M}(n)$ and defining the true community sets as

$$\Omega_i := \{v \in [n] : X_v = i\} \quad \text{for each } i \in [k],$$

the following holds:

$$\mathbb{P}\left(\max_{\pi \in S_k} \frac{1}{k} \sum_{i=1}^{k} \frac{\left| \left\{ v \in \Omega_i : \pi(\widehat{X}_v) = i \right\} \right|}{|\Omega_i|} \geq f(n) \right) = 1 - o(1),$$

where the maximum is taken over all permutations π of the k labels (to account for label symmetry).

We say that an algorithm achieves **weak recovery** if it recovers communities with accuracy at least $1/k + \Omega(1)$.

The following theorem provides a necessary and sufficient condition for weak recovery.

Theorem 3.4 (cf. [2])
Let $X, \quad \hat{X} \in \{-1, +1\}^n$. Then weak recovery is solvable if and only if

$$|\langle X, \hat{X}(G) \rangle| = \Omega(1).$$

The theorem states that weak recovery holds if the scalar product of the estimated separating vector and the true separating vector is bounded away from zero (i.e., it is lower bounded by a positive constant). In other words, the covariance between the two vectors is positive, meaning that there is some correlation between them.

3.2 Graph Powers

We are now in a position to define powers of graphs, which the authors of the paper introduced in two different ways.

Definition 3.5

Let G be a graph and let $r \in \mathbb{N}^+$ be a positive integer. The r-*th power* of G, denoted by $G^{(r)}$, is the graph with the same vertex set as G, in which two vertices are adjacent if there exists a path of length at most r between them in G.

In other words, $G^{(r)}$ contains all edges of G and adds an edge between any pair of vertices whose distance in G is at most r.

Note: One may equivalently define $G^{(r)}$ using *walks* of length at most r, instead of paths, since both lead to the same adjacency structure in this context.

Adjacency Matrix Formulation:

Let A be the adjacency matrix of the graph G, modified such that all diagonal entries are 1 (i.e., each vertex has a self-loop). Then, the adjacency matrix $A^{(r)}$ of the powered graph $G^{(r)}$ is defined as:

$$A^{(r)} = \mathbf{1}_{\{A^r \geq 1\}}$$

where

- A^r denotes the r-th matrix power of A, computed via standard matrix multiplication,
- $\mathbf{1}_{\{A^r \geq 1\}}$ is a binary matrix whose entries are 1 if the corresponding entry in A^r is at least 1, and 0 otherwise.

Thus, $A^{(r)}$ encodes all edges between vertex pairs that are connected by a walk of length at most r in G.

Our next theorem concerns the eigenvalues of the powered graph's adjacency matrix and its spectral gap.

Theorem 3.6 – Spectral separation for the distance matrix (cf. [2])
Let (X,G) be drawn from $\mathrm{SBM}(n,a,b)$ with $(a+b)/2 > 1$. Let $A^{[r]}$ be the r-distance matrix of G (i.e., $A^{[r]}_{ij} = 1$ if and only if $d_G(i,j) = r$), and let $r = \varepsilon \log(n)$ such that $\varepsilon > 0$, and $\varepsilon \log\left(\frac{a+b}{2}\right) < \frac{1}{4}$.
Then, with high probability:

A. If
$$\frac{a+b}{2} < \left(\frac{a-b}{2}\right)^2,$$
then
$$\lambda_1\left(A^{[r]}\right) \asymp \left(\frac{a+b}{2}\right)^r,$$
$$\lambda_2\left(A^{[r]}\right) \asymp \left(\frac{a-b}{2}\right)^r,$$
$$\left|\lambda_3\left(A^{[r]}\right)\right| \le \left(\frac{a+b}{2}\right)^{r/2} \log(n)^{O(1)}.$$

B. If
$$\frac{a+b}{2} > \left(\frac{a-b}{2}\right)^2,$$
then
$$\lambda_1\left(A^{[r]}\right) \asymp \left(\frac{a+b}{2}\right)^r,$$
$$\left|\lambda_2\left(A^{[r]}\right)\right| \le \left(\frac{a+b}{2}\right)^{r/2} \log(n)^{O(1)}.$$

Furthermore, let $\varphi_2(A^{[r]})$ denote the eigenvector corresponding to the second largest eigenvalue $\lambda_2(A^{[r]})$. Let the estimated labels $\hat{X} \in \{-1,+1\}^n$ be obtained via the rounding procedure:
$$\hat{X}_i = \mathrm{sign}\left(\left[\varphi_2(A^{[r]})\right]_i\right).$$
Then $\hat{X}$ achieves weak recovery whenever
$$\frac{a+b}{2} < \left(\frac{a-b}{2}\right)^2.$$

Our previous theorem, which is based on spectral methods, also has several intuitive interpretations. The fact that the first (dominant) eigenvalue is large indicates that the graph exhibits a strong community structure, making it easier to identify the communities using spectral methods. Another important fact is that the difference between the first two eigenvalues (the spectral gap) is also not small. This is good news for us, as it means that the spectral characteristics of the communities are easier to detect, since the graph structure is more robust to noise and random perturbations. In such cases, spectral algorithms tend to perform better as well. Moreover, the fact that the third (and hence the remaining) eigenvalues do not stand out implies that there are no significant substructures in the graph. Since there are only a few dominant eigenvalues, the structure of the graph is simpler and more interpretable from the perspective of community detection.

Note. A similar theorem holds not only for the distance matrix but also for the adjacency matrix of the powered graph (see [2]).

The following theorem illustrates the relationship between effective density and weak recovery, where powers of graphs appear again. By the way, the Weyl inequality also plays an important role in the proof of this theorem.

Theorem 3.7 (cf. [2])
Let $\{G_n\}_{n\geq 1}$ be a sequence of graphs such that $\operatorname{diam}(G_n) = \omega(1)$, and $\{r_n\}_{n\geq 1}$ a sequence of positive integers such that $r_n = \varepsilon \cdot \operatorname{diam}(G_n)$. Then,

$$\lambda_2(G_n^{(r_n)}) \geq (1 - o_\varepsilon(1))(r_n + 1)\hat{d}_{r_n}^{r_n/2}(G_n),$$

where

$$\hat{d}_r(G) = \left(\frac{1}{r+1} \sum_{i=0}^{r} \sqrt{\delta^{(i)}(G) \cdot \delta^{(r-i)}(G)} \right)^{2/r},$$

and

$$\delta^{(i)}(G) = \min_{(x,y)\in E(G)} |\{v \in V(G) : d_G(x,v) = i,\ d_G(y,v) \geq i\}|.$$

Note. A graph's diameter is the length of the longest shortest path between any pair of vertices in the graph. We write $f(n) = \omega(g(n))$ to denote that

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty.$$

We therefore assume that the diameters of the graphs tend to infinity.

In the theorem, for a graph G, the quantity $\delta^{(i)}(G)$ is defined as the minimum, taken over all edges $(x,y) \in E(G)$, of the number of vertices $v \in V(G)$ such that the shortest path from v to x has length exactly i, and every path from v to y has length at least i. So the quantity $\delta^{(i)}(G)$ measures, for a given edge, how many vertices "see" one endpoint better than the other at distance i. The quantity $\hat{d}_r(G)$ averages this information symmetrically: for each $i \in \{0,1,\ldots,r\}$, it takes the square root of the product $\delta^{(i)}(G) \cdot \delta^{(r-i)}(G)$, then averages over all such i and normalizes the result using the r-th root. Intuitively, $\hat{d}_r(G)$ measures how well vertices can be distinguished from each other in the r-step neighborhoods around edges. The larger $\hat{d}_r(G)$ is, the more vertices exist that are symmetrically distinguishable from the two endpoints of an edge at some distance scale. This indicates that the graph contains more structured information in its r-step local environment, which in turn supports spectral separation and enhances the robustness of spectral algorithms. As shown in the theorem, the second eigenvalue becomes large when the spectral density is also large.

Summary

This thesis explored the behavior of singular and eigenvectors in random matrix models under various perturbation settings and examined their applications in statistical problems, particularly community detection. In the first part, we focused on the theoretical underpinnings of singular value decomposition (SVD), highlighting how the addition of random noise — such as Bernoulli or Gaussian perturbations — affects the stability of low-rank approximations and the geometry of singular vectors. We presented relevant perturbation bounds and supported them with numerical simulations.

In the second part, we turned to the Stochastic Block Model (SBM), a key framework in modern network analysis. We studied spectral methods for reconstructing latent group structures and analyzed the robustness of these techniques when the observed adjacency matrix is corrupted by random noise. We also investigated the relationship between SBM and the $\mathbb{Z}_2$ synchronization problem, showing how different noise models affect classification accuracy. Both synthetic and real-world data examples were presented to illustrate the practical implications of theoretical guarantees.

Finally, we extended our analysis to sparse graphs and adversarial settings by leveraging powers of graphs. We demonstrated that graph powering enhances spectral separation and enables weak recovery even when a bounded number of adversarial edge modifications are allowed. Our simulations confirmed that spectral methods remain effective under these more challenging circumstances.

In summary, the results of this thesis contribute to a deeper understanding of how spectral algorithms behave under randomness and noise, offering insights into when reliable reconstruction is possible and how to design robust techniques for large-scale data analysis.

Bibliography

[1] Emmanuel Abbe, Jianqing Fan, Kaizheng Wang, Yiqiao Zhong: *Entrywise eigenvector analysis of random matrices with low expected rank*, Ann Stat. 48(3):1452–1474 (2020) (https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8046180/)

[2] Emmanuel Abbe, Enric Boix, Peter Ralli, Colin Sandon: *Graph powering and spectral robustness*, arXiv:1809.04818 (Sep. 2018) (https://arxiv.org/abs/1809.04818)

[3] Bader Al-Anzi, Sherif Gerges, Noah Olsman, Christopher Ormerod, Georgios Piliouras, John Ormerod, Kai Zinn: *Modeling and analysis of modular structure in diverse biological networks*, J. Neurosci. Methods (June 2017) (https://www.sciencedirect.com/science/article/pii/S0022519317301534)

[4] Arash Amini, Marina Paez, Lizhen Lin: *Hierarchical Stochastic Block Model for Community Detection in Multiplex Networks*, Bayesian Anal. (March 2024) (https://doi.org/10.1214/22-BA1355)

[5] Benaych-Georges, F.; Nadakuditi, R. R.: *The singular values and vectors of low rank perturbations of large rectangular random matrices*, J. Multivariate Anal. 111 (2012), 120–135. (https://arxiv.org/abs/1103.2221)

[6] Blum, A.; Hopcroft, J.; Kannan, R.: *Foundations of Data Science*, Cambridge University Press (2020) (https://doi.org/10.1017/9781108755528, cf. https://www.cs.cmu.edu/~venkatg/teaching/CStheory-infoage/book-chapter-4.pdf)

[7] Chen, Y.; Cheng, Ch..; Fan, Y.: *Asymmetry helps: Eigenvalue and eigenvector analyses of asymmetrically perturbed low-rank matrices*, Ann. Stat. 49(1):435 (2021) (https://doi.org/10.1214/20-aos1963, cf. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8300484/pdf/nihms-1639565.pdf)

[8] Patrick Doreian, Vladimir Batagelj, Anuška Ferligoj: *Bayesian Stochastic Blockmodeling*, Adv. Netw. Clust. Blockmodel. (Nov. 2019)
(https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119483298.ch11)

[9] Brenden P. Epps, Erik M. Krivitzky: *Singular value decomposition of noisy data: noise filtering*, Flow Turbul. Combust. 60 (2019)
(https://link.springer.com/article/10.1007/s00348-019-2768-4)

[10] Fan, Jianqing; Wang, Weichen; Zhong, Yiqiao: *An l^∞ eigenvector perturbation bound and its application to robust covariance*, J. Mach. Learn. Res. 18 (2017), Paper No. 207, 42 pp.
(https://www.jmlr.org/papers/volume18/16-140/16-140.pdf)

[11] Mario Frank, Joachim M. Buhmann: *Selecting the rank of truncated SVD by Maximum Approximation Capacity*, Proc. IEEE. (2011)
(https://www.researchgate.net/publication/224260379_Selecting_the_rank_of_truncated_SVD_by_maximum_approximation_capacity)

[12] Lennart Gulikers, Marc Lelarge, Laurent Massoulié: *An impossibility result for reconstruction in the degree-corrected stochastic block model*, Ann. Appl. Probab. (Oct. 2018)
(https://doi.org/10.1214/18-AAP1381)

[13] Daniel Gribel, Thibaut Vidal, Michel Gendreau: *Assortative-Constrained Stochastic Block Models*, Proc. IEEE. (Jan. 2021)
(https://arxiv.org/abs/2004.11890)

[14] Hamid Hassanpour, Mostefa Mesbah, Boualem Boashash: *Time-Frequency Feature Extraction of Newborn EEG Seizure Using SVD-Based Techniques*, J. Appl. Math. (Dec. 2004); Article ID 898124.
(https://link.springer.com/article/10.1155/S1110865704406167)

[15] Jong Min Lim, Christopher L. DeMarco: *SVD-Based Voltage Stability Assessment From Phasor Measurement Unit Data*, Proc. IEEE. (July 2016)
(https://ieeexplore.ieee.org/document/7302608)

[16] Cathy Xuanchi Liu, Tristram J Alexander, Eduardo G Altmann: *Nonassortative relationships between groups of nodes are typical in complex networks*, PNAS Nexus (Nov. 2023)
(https://academic.oup.com/pnasnexus/article/2/11/pgad364/7367864)

[17] O'Rourke, S.; Vu, V.; Wang, K.: *Random perturbation of low rank matrices: Improving classical bounds*, Linear Algebra Appl. 540 (2016), 26–59.

[18] T. N. Palmer, R. Gelaro, J. Barkmeijer, R. Buizza: *Singular Vectors, Metrics, and Adaptive Observations*, J. Atmos. Sci. 55 (1998): 633–653.
(https://journals.ametsoc.org/view/journals/atsc/55/4/1520-0469_1998_055_0633_svmaao_2.0.co_2.xml)

[19] P. A. Ramachandran: *Method of fundamental solutions: singular valuedecomposition analysis*, Commun. Numer. Methods Eng. 18 (2002): 789–801.
(https://onlinelibrary.wiley.com/doi/epdf/10.1002/cnm.537)

[20] Nicholas D. Sidiropoulos, Lieven De Lathauwer: *Tensor Decomposition for Signal Processing and Machine Learning*, Proc. IEEE. (July 2017)
(https://arxiv.org/abs/1607.01668)

[21] Ludovic Stephan, Laurent Massoulié: *Robustness of spectral methods for community detection*, arXivSBM (Nov. 2018)
(https://arxiv.org/abs/1811.05808)

[22] Air traffic control
(http://konect.cc/networks/maayan-faa/)

[23] Weyl's inequality
(https://en.wikipedia.org/wiki/Weyl%27s_inequality)

NYILTAKOZAT

Alulírott **Kovács Sebestyén** nyilatkozom, hogy szakdolgozatom elkészítése során az alább felsorolt feladatok elvégzésére a megadott MI alapú eszközöket alkalmaztam:

Feladat	**Felhasznált eszköz**	**Felhasználás helye**	**Megjegyzés**
Irodalomkeresés	–	–	–
Ábra készítés	GPT-4o	az ábrák többsége	Python kód generálás
Adatfeldolgozás	–	–	–
Szövegvázlat készítés	GPT-4o	1. fejezet, 4. oldal 2. fejezet, 16. oldal	SVD és SBM gyakorlati alkalmazásainak felsorolása
Nyelvhelyesség ellenőrzése	–	–	–

A felsoroltakon túl más MI alapú eszközt nem használtam.
Budapest, 2025. május 31.

Kovács Sebestyén