Név: **Kovács Sebestyén**
NEPTUN-kód: **FP959I**
Témavezető: **Backhausz Ágnes Mariann**

Math project 3
Random matrices, perturbations and their applications in statistics

---

1 Introduction

The problem of perturbation plays a significant role in statistical modeling. Often, certain data can only be observed with noise, while we are interested in the actual data series. In previous semesters, we examined specific theorems to see how eigenvectors (or singular vectors) could change under small modifications and how sensitive they are to these changes. In previous semesters, we also studied theorems related to such modifications, and using computer simulation, we examined whether the theorems remain valid when certain conditions are modified. These theorems came from the articles [5] and [6]. I mainly focused on the theorems in practice that describe the changes in eigenvectors during perturbation. I explain how this semester's work relates to the previous ones in more detail in Section 5. I would like to note, though, that if the noise matrix in the perturbation is a Bernoulli matrix (with ± 1 entries), the new eigenvectors will not exhibit extreme behavior. I also examined this for the Wishart matrix, and I did not observe such behavior here neither. However, this semester, we also investigated concrete applications within the stochastic block model, where we aimed to identify clusters (denser groups) in graphs using eigenvectors. We also tested the algorithm of this model on real-world data. Another connection to the previous semesters is that often we can only observe the graph's adjacency matrix with noise. This semester I read through the paper by Emmanuel Abbe, Jianqing Fan, Kaizheng Wang, and Yiqiao Zhong ([1]), whose main topic is the presentation and solution of the problem related to the Stochastic Block Model. Now I would like to briefly explain the Stochastic Block Model.

The Stochastic Block Model (SBM) is a probabilistic model used in network analysis, particularly for identifying community structures within a graph. It assumes that a network is composed of different groups or "blocks" (communities), and the probability of an edge (connection) between two nodes depends only on the group memberships of those nodes. It is important to draw the edges independently from one another. We typically assume that there are two groups, with different probabilities for edges to occur within the groups and between the groups. In general, edges are denser within groups, so our goal is to reconstruct these denser groups from the random graph, given that we only observe the edges. The SBM is particularly useful for modeling the structure of social groups, sub-networks, and communities, as it takes into account the probabilistic distribution of connections between nodes, which reflects patterns of social interactions or relationships. This helps uncover hidden patterns and relational structures in social networks.

2 Simulations for community detection

Now I would like to briefly explain the algorithm written in the article for solving the community detection problem in the Stochastic Block Model. Let us 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 to the second group ($J = V/I$). We will estimate this vector with $\hat{x} \in \{1,-1\}^n$ with small possible error. The algorithm is really easy: we need to calculate 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 $\hat{x}^i := \operatorname{sgn}(u_2^i)$.

The article specifically emphasizes that the coordinates of the second eigenvector also provide information about the quality of separation. If the clusters are well-separated, the coordinates of the second eigenvector can be clearly divided into two groups. This theorem states that we obtain a good estimate:

Theorem 1. (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:

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

The theorem 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 $\hat{x}$:

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

I tested the algorithm written in the article in a specific case with a fixed number of vertices. I 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. I 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 1 and in figure 2

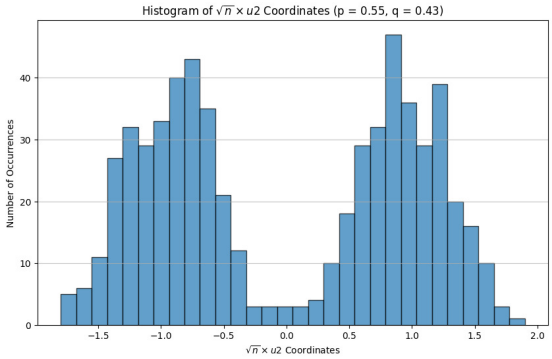

Figure 1: Histogram of $\sqrt{n}$ coordinates $(p = 0.55, q = 0.43)$

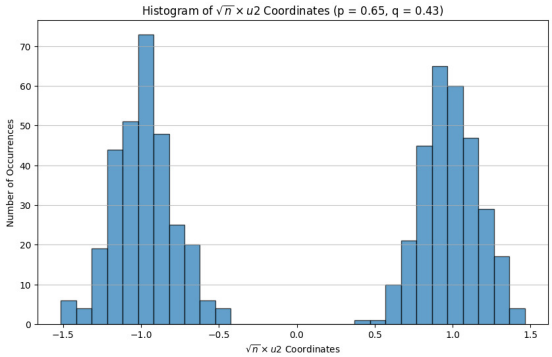

Figure 2: Histogram of $\sqrt{n}$ coordinates $(p = 0.65, q = 0.43)$

I 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 I did not change the 8 cases, nor did I alter the graphs. I 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^2distance between the separating and estimated vector can be estimated by averaging such observations:

$$\|x - \hat{x}\|_2 = \sqrt{\sum_{i=1}^{n} (x_i - \hat{x}_i)^2}.$$

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 3: The average of the missclassification rates

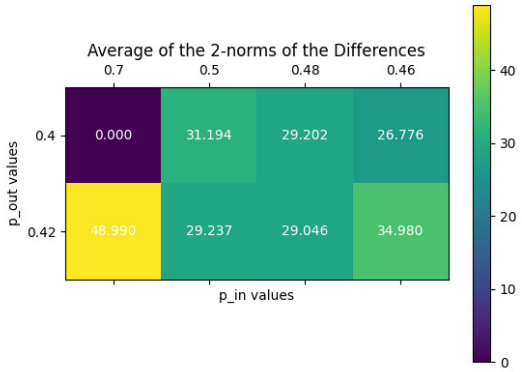

Figure 4: The average of the 2-norms of the differences

Figure 3 and figure 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.

3 $\mathbb{Z}_2$ synchronization

The $\mathbb{Z}_2$ synchronization problem is very similar to the stochastic block model task. In the latter, the goal is to cluster the vertices of a graph into groups using random edges. In the $\mathbb{Z}_2$ synchronization problem, we observe noisy versions of random ± 1 values and aim to filter out the noise, which is typically chosen from a normal distribution.

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

$$Y_{ij} = x_i \cdot x_j + \sigma \cdot W_{ij} \quad (\text{where } x \in \{\pm 1\}^n, \quad i < j \Rightarrow W_{ij} \sim N(0,1), \quad \sigma > 0, \quad W_{ii} = 0 \quad \text{and} \quad W_{ij} = W_{ji}.)$$

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 $\sigma = \sqrt{\frac{n}{2\log n}}$, exact recovery is achievable for noise levels smaller than this.

Theorem 2. (cf. [1]) We suppose that $\sigma \leq \sqrt{\frac{n}{(2+\varepsilon)\log n}}$ for some $\varepsilon > 0$.With probability $1 - o(1)$, the leading eigenvector u of Y with unit ℓ_2 norm satisfies

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

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

According to the theorem, 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.

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

Figure 5 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, I applied the stochastic block model algorithm to the noisy adjacency matrix and evaluated the accuracy of the reconstruction with the added noise. For my 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. I 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 6. If I had used the information-theoretic threshold from the $\mathbb{Z}_2$ synchronization

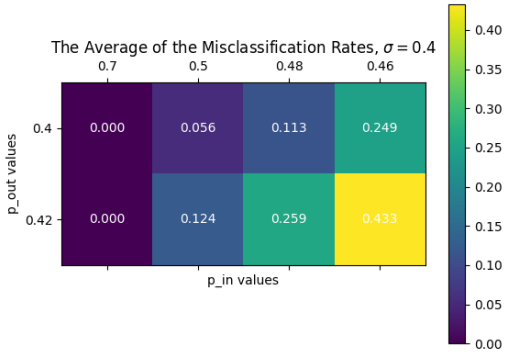

Figure 6: The average misclassification rates with noise $\sigma = 0.4$

theorem as the noise level, the misclassification rate would have gone up to 45%. This is because, in that theorem, noise is added to $x^\top x$ not directly to the adjacency matrix. (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.)

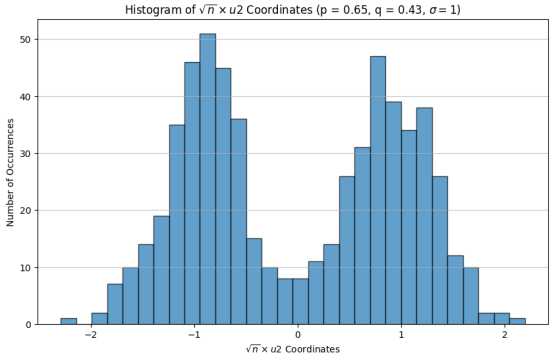

Figure 7: The histogram of the coordinates of $\sqrt{n} \cdot u_2$ with $p_{in} = 0.65$, $p_{out} = 0.43$, $\sigma = 1$

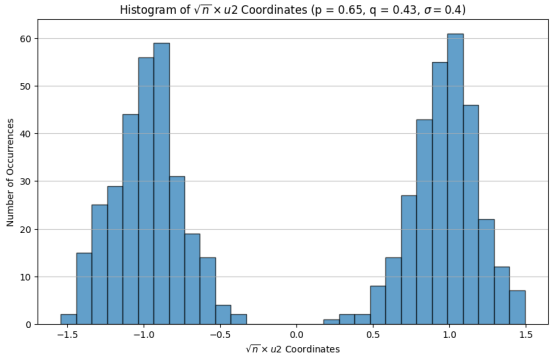

Figure 8: The histogram of the coordinates of $\sqrt{n} \cdot u_2$ with $p_{in} = 0.65$, $p_{out} = 0.43$, $\sigma = 0.4$

In figure 7 and figure 8 considered the same problem, calculated the second eigenvector of the noisy version of the adjacency matrix, and assigned the vertices to groups based on its 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 ($\sigma = 0.4$).

4 Application on real data

I tested the stochastic block model algorithm on a deterministic graph as well, where the edge between two vertices does not depend on randomness. I downloaded the graph from [7]. I 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 and clustering coefficient.

Definition. 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.

Definition. 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 I ran the stochastic block model. The graph and the two groups generated by the model are depicted in 9. We expect higher edge density and clustering coefficient within the groups.

The descreption 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

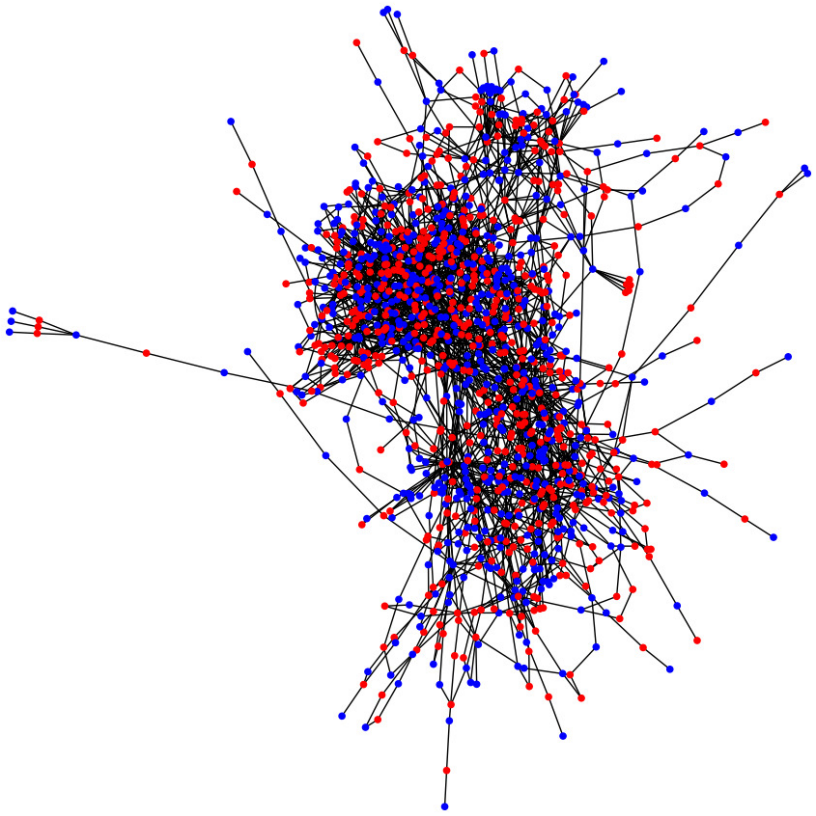

Figure 9: The two groups of our graph

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 the other three 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 10. This means that randomness plays a significant role in whether the two groups are sufficiently separated from each other.

5 Connection with the results of the previous semesters

Our third semester project work is quite closely related to the previous two semester projects. In earlier projects, we examined how adding random noise to a random matrix affects the eigenvectors (singular

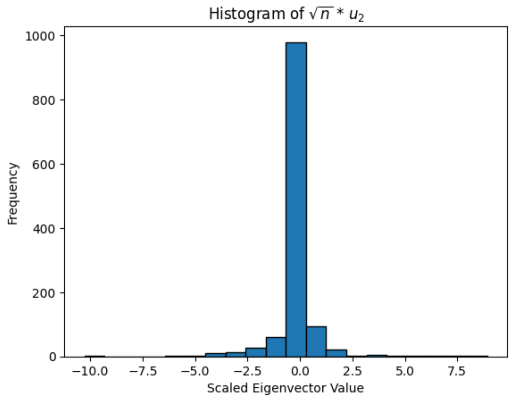

Figure 10: The coordinates of the second eigenvector, real data

vectors), which are effective in describing the properties of the system. In the first semester task, we focused on investigating the changes in the first singular vector:

Theorem 3. (cf. [6]) 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).

This theorem is related to the phenomenon observed in the stochastic block model. There, the eigenvector of the random graph's adjacency matrix had to be calculated, and the component-wise signs of this eigenvector served as the separating vector. Interestingly, in the SBM, the second eigenvector played a prominent role. We also tested this algorithm in the third point, where we added random noise to the random adjacency matrix and calculated the sign of the second eigenvector of this new matrix. The common element in both phenomena discussed is the addition of noise, but in the first semester, we investigated the change in the first eigenvector, not the second, and the noise was not normally distributed but rather a Bernoulli matrix. In the second semester's work, we also examined the changes in the singular vectors. In that case, we first constructed a matrix from the singular vectors of the original system as column vectors. Then we subtracted the matrix formed by the singular vectors of the system observed with noise from this original matrix. The norm of the resulting error matrix could be estimated from above by a constant multiplier for the infinity norm of the noise matrix. The noise matrix here was also a Bernoulli matrix. This theorem is stated as follows:

Theorem 4. (cf. [5]) We suppose that $\delta > \|E\|_2$ and $\sigma_r - \varepsilon = \Omega(r^3\mu^2\|E\|_\infty)$, where $\varepsilon = \|A - A_r\|_\infty$. If A is symmetric and for any $i = 1, \ldots r$ the interval $[\sigma_i - \delta, \sigma_i + \delta]$ does not contain any singular values of A other than σ_i, then

$$\|V' - V\|_{\max} = \mathcal{O}\left(\frac{r^4\mu^2\|E\|_\infty}{(\sigma_r - \varepsilon)\sqrt{n}} + \frac{r^{\frac{3}{2}}\mu^{\frac{1}{2}}\|E\|_\infty}{\delta\sqrt{n}}\right).$$

We also conducted a simulation for this theorem, where the E matrix was a ± 1 Bernoulli matrix, and A was drawn from the Wishart distribution. Interestingly, according to the article, as the number of nodes increases in large graphs, the first two eigenvectors, which come from the adjacency matrix, increasingly approximate the true clustering structure. This fact is regularly applied in practice because, in large networks, the relationships and separations between clusters become increasingly important.

In my thesis, I would like to continue exploring the phenomena discussed so far. I plan to delve deeper into the literature and apply the models to larger real-world datasets.

References

[1] Emmanuel Abbe, Jianqing Fan, Kaizheng Wang, Yiqiao Zhong: *Entrywise eigenvector analysis of random matrices with low expected rank*, Ann Stat. 2020 June ; 48(3): 1452–1474. doi:10.1214/19-aos1854.
(https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8046180/)

[2] 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.

[3] 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)

[4] Chen, Y.; Cheng, Ch..; Fan, Y.: *Asymmetry helps: Eigenvalue and eigenvector analyses of asymmetrically perturbed low-rank matrices*, Annals of statistics, 2021, 49.1: 435.
(https://doi.org/10.1214/20-aos1963, cf. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8300484/pdf/nihms-1639565.pdf)

[5] Jianqing Fan, Kaizheng Wang, Yiqiao Zhong: *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)

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

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