Exploring the Fundamental Connections between Network Science and Graph Signal Processing

Heng-Chien Liou*, Wen-Chiao Tsai†, Yu-Cheng Lin‡
Dept. Electrical Engineering
National Taiwan University Taipei, Taiwan
*b05901023@ntu.edu.tw, †b05901065@ntu.edu.tw, ‡b05502145@ntu.edu.tw

***Abstract*—This report is for the final project of EE3018 Electrical Engineering Lab (topics on Communication System), in 2019 Spring, National Taiwan University. Several graph signal processing tools and concepts are utilized to study well-known network science topics. One spectral clustering algorithm is implemented and compared to principal component analysis.**

I. Introduction

Graph signal processing is an emerging field to solve the problem of signal in irregular domain, that is, graph or network [1], [2]. On the other hand network science is a field started growing at the end of last century [3], [4], with wide range of applications in various fields, such as sociology [5], biology [6], and psychology [7]. Both fields consider what happens on graph, one focuses on the mathematical foundation and signal processing tool, and other focuses on the dynamics, evolution and equilibria of network process. Using a metaphor, graph signal processing is the signal processing on graph, and network science is the statistical physics on graph [8].

Although the two fields seems disjoint at first glance, there are actually many common interests underlying. For example, both graph theory and spectral graph theory [9]–[11] have major rules in the foundation of both fields.

In this research, we used some simple graph signal processing tool to analyze several network science topics. Specifically, three topics will be addressed. The first one is the topology of network, most of the time this is related to network model, some-times called the dynamics of network [12]. The second one is the change of nodes' state on network, such as epidemic [13], this topic is also called dynamics on network [12]. At the end, we implement some algorithm for network science problem developed from graph signal processing concept.

A. Experiment Environment

Most of the experiments are carried out in Python3 [14] programming language, with package PyGSP [15], which is developed for signal processing on graph. In some experiments, package NetworkX [16] is also used for network manipulation. Other relevant packages are NumPy [17], Matplotlib [18], seaborn [19], scikit-learn [20], keras [21].

Manuscript created June 24, 2019; revised June 18, 2020. We appreciate Prof. Liu for advice and reviews regarding this project.

II. Network Topology

In this section, we denotes N as number of nodes, N_e as number of edges.

Also note that, for most of the visualization about eigenvalue, the rug plots (marks on axis) display the experimental results of the eigenvalue's location. The density plot generated by seaborn [19] is intended to help to intuitively observed how eigenvalue distributed, without actual empirical meaning.

A. Erdös-Rényi Model

In this subsection we explore the Erdös-Rényi Model [22], which is arguably the most simple network model. We first tried Erdös-Rényi graph (ER graph for short) with $N = 100$, $p = 0.5$. The corresponding distribution of eigenvalue is shown in Fig. 1.

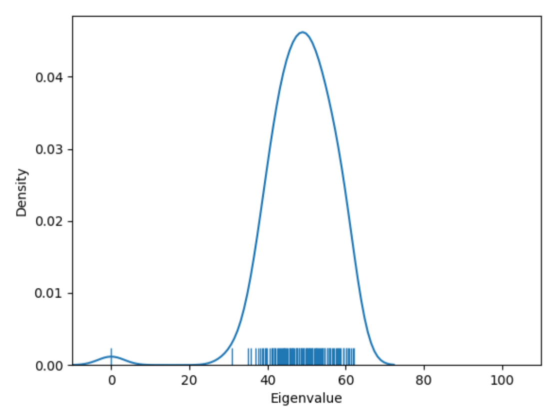

Fig. 1. Erdös-Rényi graph with $N = 100$, $p = 0.5$ and $N_e = 2433$

If looked closely, it can be observed that most of the eigenvalue is centered around 50, which is equal to $N \cdot p = 100 \cdot 0.5 = 50$, in this case. Suspecting this is not coincidence, we carried out two more examination with $p = 0.2$ and $p = 0.8$, as shown in Fig. 2 and 3 respectively. And the result, as shown, justify the speculation that the distribution of eigenvalue will center around $N \cdot p$.

B. Barabási-Albert Model

And then we explore one of the most influential network model, Barabási-Albert Model (BA model).

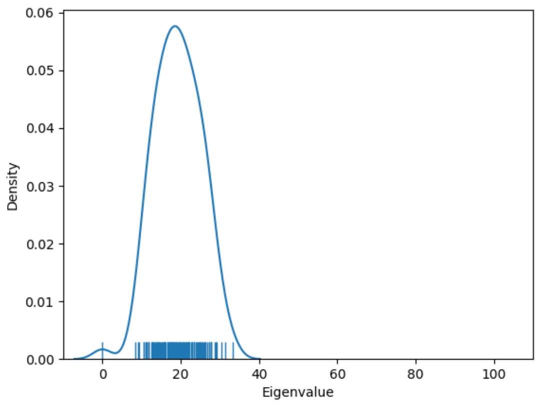

Fig. 2. Erdös-Rényi graph with $N = 100$, $p = 0.2$, $N_e = 964$

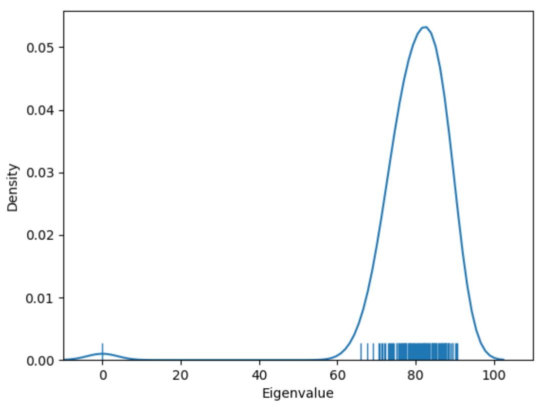

Fig. 3. Erdös-Rényi graph with $N = 100$, $p = 0.8$, $N_e = 4004$

First we generate a network with $N = 100$, $m_0 = 5$ and $m = 5$. The resulting eigenvalue distribution is shown in Fig. 4. The resulting graph has number of edges$= 475$.

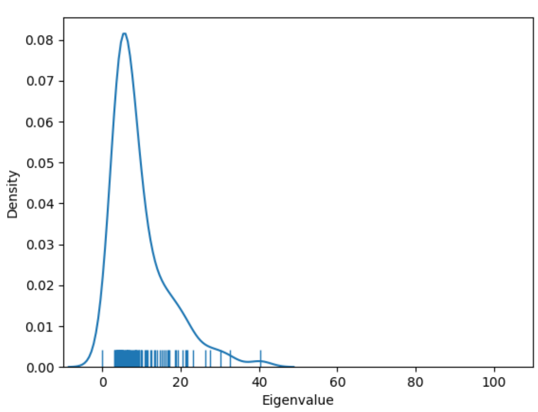

Fig. 4. Barabási-Albert graph with $N = 100$, $m_0 = 5$, $m = 5$ and$N_e = 475$

In Section II-A, one can argue that the difference in eigenvalue distribution only comes from the difference in number of edges. To get rid of this possibility, we also generate an Erdös-Rényi graph with $N = 100$, $p = 0.1$, the resulting edge is $N_e = 462$, which is comparable to the graph in Fig. 4. The result is shown in Fig. 5

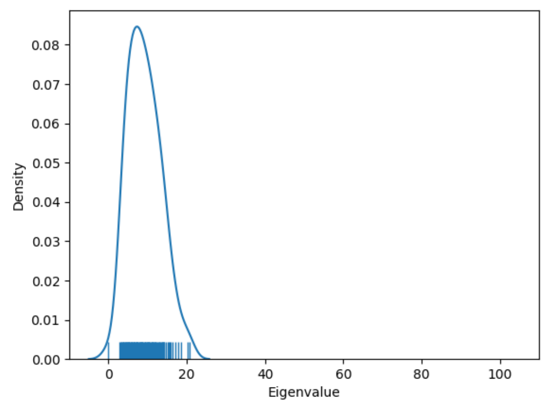

Fig. 5. Erdös-Rényi graph with $N = 100$, $p = 0.1$, $N_e = 462$

Comparing Fig. 4 and Fig. 5, the difference is noticeable. In Fig. 5 Erdös-Rényi model generates a bell-shape-like distribution of eigenvalue, while in Fig. 4 Barabási-Albert model generates a distribution with longer tail. The long tail for BA model in eigenvalue distribution is kind of analogy of scale-free property in degree distribution [8].

C. *Watts-Strogatz model*

And we move our focus to Watts-Strogatz model [23], the so-called "Small World Model".

First, we observed the extreme case where rewiring probability $p = 1$. The result is shown in Fig. 6. We can observe that the result is very similar to Fig. 2, since Watts-Strogatz model with rewiring probability $p = 1$ will be reduced to completely random graph [23].

We also examined the other two cases where $p = 0$ and $p = 0.5$, the result is shown in Fig. 7 and Fig. 8 respectively.

D. *Ring Graph*

At the end of this section, we went back to the one of the most fundamental graph, ring graph.

The result with $N = 100$ is show in Fig. 9. We can observe that the eigenvalues are very concentrated. Thus, we focus on smaller range of eigenvalue axis, the result is shown in Fig. 10. We can observe that the eigenvalue is centered around value 2.

While there is analogy for Laplacian operator in vertex domain and Euclidean space [1], however, directly calculate the Laplacian of ring graph will not lead to equally spaced square rooted eigenvalue, which is the case in digital signal processing. This might seems like a counter-evidence of equivalence of graph signal processing on ring graph and digital signal processing.

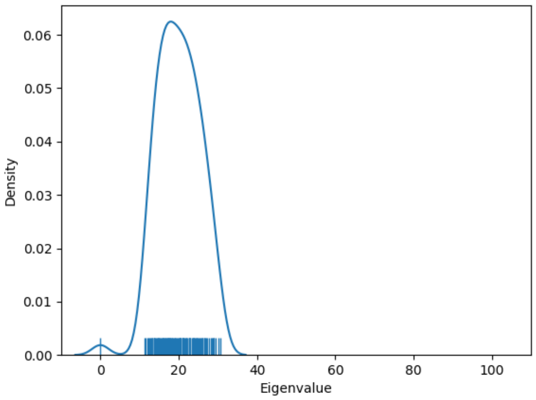

Fig. 6. Watts-Strogatz model with $N = 100$, $k = 20$, $p = 1$

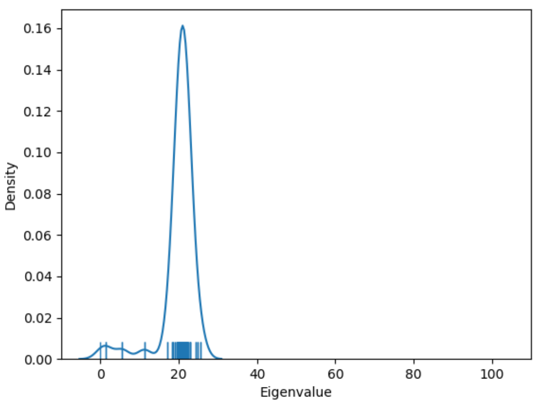

Fig. 7. Watts-Strogatz model with $N = 100$, $k = 20$, $p = 0$

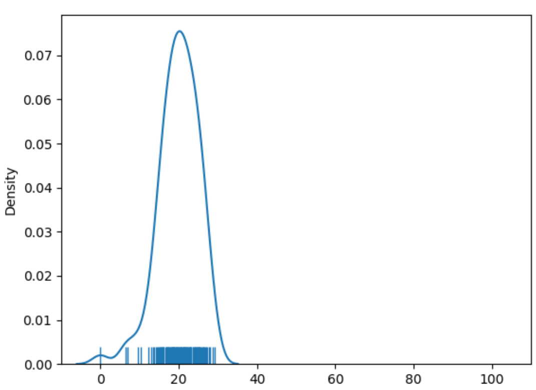

Fig. 8. Watts-Strogatz model with $N = 100$, $k = 20$, $p = 0.3$

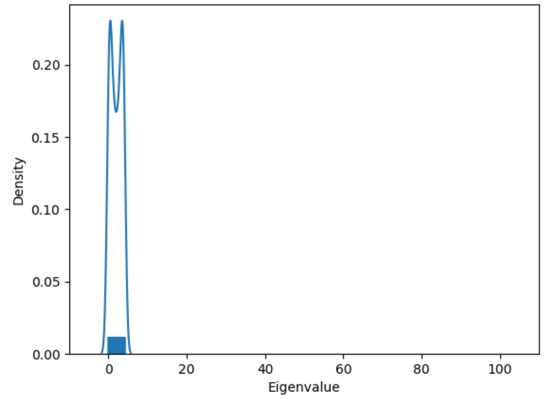

Fig. 9. Ring Graph with $N = 100$

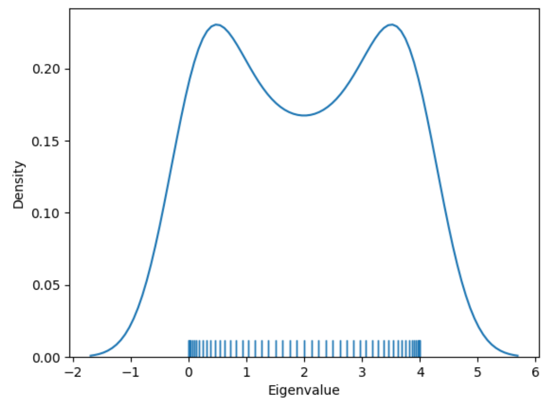

Fig. 10. Ring Graph with $N = 100$, while focus in the range around concentration.

Indeed, as indicated in [24], the eigenvalues are actually $2 - 2\cos(2\pi k/n)$. Thus, we minus the eigenvalue by 2, and take $\cos^{-1}$. The results are indeed equally spaced as in Fig. 11.

E. Conclusion

We concluded our experiment result about network topology and corresponding eigenvalue pattern.

- The eigenvalues of Laplacian do not only capture the information about the number of edges in network, this is proven by the comparison between Fig. 4 and Fig. 5.
- The eigenvalue of Laplacian seems to have correlation with degree distribution [3] of the given networks. For example, the eigenvalue in Fig. 4 resembles the scale-free [4], [25], [26] property of Barabási-Albert Model; and the eigenvalues in Fig. 9 are highly concentrated around value 2, while all nodes in ring graph have degree equals to 2.

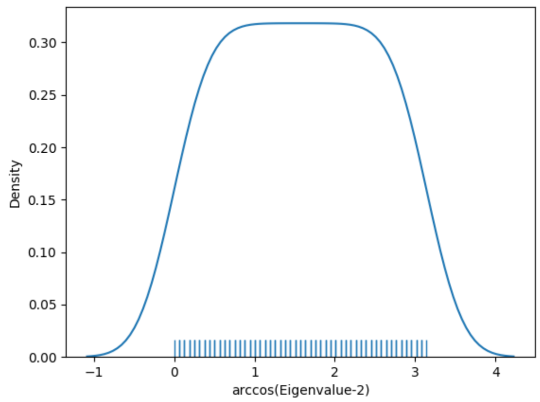

Fig. 11. Ring Graph with $N = 100$, while minus by 2, and taking inverse cosine operation.

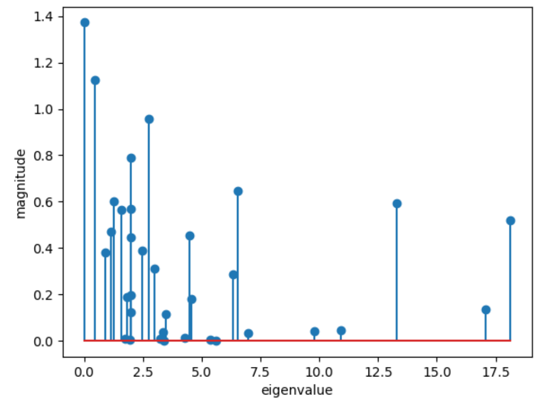

Fig. 12. The result of Graph Fourier Transform of snapshot of epidemic

F. Future Direction

We discussed some possible future direction regarding graph signal processing and network topology.

1) Probability distribution of eigenvalue pattern: If we regard the associated eigenvalue pattern for a given network as a vector, then for every network model, we can also regard the eigenvalue pattern as a random vector.

In other worlds, a probabilistic process, network model with given parameter, produce a network $\mathcal{X}$, and let the eigenvalue vector associate with $\mathcal{X}$ be $\mathrm{f}(\mathcal{X})$. Since $\mathcal{X}$ is a random object from a sample space, $\mathrm{f}(\mathcal{X})$ will also be a random vector.

How to determine the probability distribution of eigenvalue pattern, by using probabilistic analysis or empirical simulation, might be an interesting questions.

III. Dynamics on Networks

In this section, we consider the dynamics on network as graph signal. And use graph signal processing tool to analyze.

A. Epidemic

Epidemic process in network has been the center of network research for years. There are several epidemic model available, such as SI, SIS, SIR [13] and even more realistic model where networks change adaptively regarding to epidemic spreading [27].

For simplicity, we first utilized SI model. We treated infected node as having graph signal 1, other susceptible nodes have signal 0.

The network data used is the well-known Zachary's Karate Club [28]. The initial infected node is randomly selected.

We snapshot the graph signal and perform further measure, the number of infected nodes is 8.

The result of Graph Fourier Transform (GFT) is shown in Fig. 12. We also calculate the high energy concentration ratio ($ECRH_\alpha$) and low energy concentration ratio ($ECRL_\gamma$) proposed in [29]. With $\alpha = 0.3$ and $\gamma = 0.3$, the result is $ECRH_{0.3} = 0.17$ and $ECRL_{0.3} = 0.47$.

B. Random Failure

For comparison purpose, we also carried out simulation on Zachary's Karate Club with failure number$= 8$ as the same as in Section III-A.

The result of Graph Fourier Transform (GFT) is shown in Fig. 13. With $\alpha = 0.3$ and $\gamma = 0.3$, the result is $ECRH_{0.3} = 0.31$ and $ECRL_{0.3} = 0.34$.

Compared to the result in Section III-A, random failure indeed concentrate more energy in high frequency and less energy in low frequency, as claimed in [29].

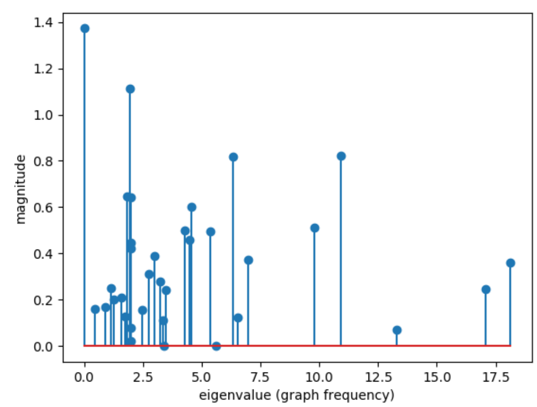

Fig. 13. The result of Graph Fourier Transform of snapshot of random failure

C. Conclusion and Discussion

Since graph signal processing methods take both graph topology and signal into consideration, graph signal processing might be a very strong candidate for network dynamics research.

There are also other researches take regards of graph spectral theory in network dynamic research. For example, [30] use eigenvalue perspective to theoretically analyze the threshold of epidemic and disinfection. [31] developed spectral approach to analyze the localization and spreading for SIS model.

IV. Application: Spectral Clustering

In this section, we implemented application using graph signal processing method to solve network science problem, specifically, we focus on the problem of community detection.

A. Problem of Community Detection

Intuitively speaking, community detection is trying to find the underlying structure in networks. For example, in social network, there are strongly cohesive structures like family, club, school friends. These communities play important role in the topology and functioning of network.

One well-known example of the power of community is the case of Zachary's karate club [28]. In this case, there were a conflict between club administrator and instructor. Later, the club split into two, one surround instructor and form a new club, other surround administrator and find a new instructor.

However, although community detection is easily to understand intuitively, it is indeed not well defined mathematically [32]. Different researcher proposed different matrices and criterion for community detection problem, and dozens of algorithms are proposed, based on optimization, information theory, dynamics and statistical inference respectively [33].

B. Using Spectral Clustering for Community Detection

One of the methods to solve community detection problem is using spectral clustering. Spectral clustering [34] only rely on the graph Laplacian matrix and can be easily implemented.

Spectral clustering takes the advantage that Laplacian matrix carry the complete information of network topology, and use eigenvectors to carry out clustering. The reason why spectral clustering works can be viewed from the point of graph cut, random walk and perturbation theory [34].

C. Comparing PCA to Spectral Clustering on Community Graph

We compared spectral clustering to principal component analysis (PCA) [35], [36]. The data is constructed using the community graph model in PyGSP [15], which can specify the number of cluster, inter- and intra-community edge density. The result is demonstrated in Fig. 14 and Fig. 15.

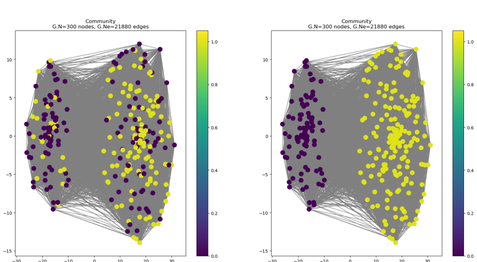

Fig. 14. Left-half is the result of PCA, and right-half is the result of spectral clustering. The node with same color is classified in same group. Number of community is 2.

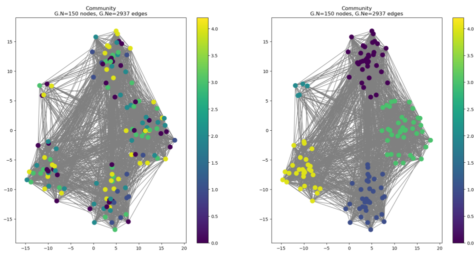

Fig. 15. Left-half is the result of PCA, and right-half is the result of spectral clustering. The node with same color is classified in same group. Number of community is 5.

D. Comparing PCA with k-means to Spectral Clustering on Image Data Clustering

We then tried to transfer these methods to different scenario, that is, we jumped out of community detection on graph to more general clustering problem in machine learning.

The data is from [37]. The task is to identify whether two image is come from same dataset, using unsupervised learning method.

First we apply autoencoder for dimensonality reduction [38]. The output of autoencoder is then pass through the principal component analysis and spectral clustering respectively. The accuracy rate is then calculated.

For PCA, after reduction, we utilize k-means method for clustering. As for spectral clustering, the graph is built using cosine-similarity as edge weight.

With data size of 5000 images, PCA with k-means reach accuracy of 0.9862, and Spectral Clustering reach accuracy of 0.9328. Both reach very high accuracy.

E. Conclusion and Discussion

Spectral clustering method outperformed principal component analysis in community detection problem on graph. On the other hand, PCA had marginal win in the clustering task of image data.

The difference might lie at, in image data clustering, we have autoencoder to help dimensionality reduction and other preprocessing. This kind of tool is not available in plain community detection on graph. And we also need to construct graph for image data clustering before utilizing spectral clustering, how we construct the graph might change the performance of spectral clustering.

V. Conclusion

In this paper, we utilized graph signal processing tool to address several topics in network science, and implemented one algorithm for community detection.

We argue that by using graph signal processing for network science research, there are many potential problem to be explored in both fields. We encouraged more researchers to engage in the intersection of these two fields.

Appendix

The source files are available upon request.

References

[1] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *arXiv preprint arXiv:1211.0053*, 2012.

[2] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proceedings of the IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[3] M. Newman, *Networks: an introduction*. Oxford university press, 2010.

[4] A.-L. Barabási *et al.*, *Network science*. Cambridge university press, 2016.

[5] D. A. McFarland, K. Lewis, and A. Goldberg, "Sociology in the era of big data: The ascent of forensic social science," *The American Sociologist*, vol. 47, no. 1, pp. 12–35, 2016.

[6] M. Gosak, R. Markovič, J. Dolenšek, M. S. Rupnik, M. Marhl, A. Stožer, and M. Perc, "Network science of biological systems at different scales: a review," *Physics of life reviews*, vol. 24, pp. 118–135, 2018.

[7] H.-C. Liou and H.-Y. Hsieh, "Modeling friendship networks among agents with personality traits," *arXiv preprint arXiv:2004.12901*, 2020.

[8] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of modern physics*, vol. 74, no. 1, p. 47, 2002.

[9] F. R. Chung and F. C. Graham, *Spectral graph theory*. American Mathematical Soc., 1997, no. 92.

[10] P. Van Mieghem, *Graph spectra for complex networks*. Cambridge University Press, 2010.

[11] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics reports*, vol. 424, no. 4-5, pp. 175–308, 2006.

[12] T. Gross and H. Sayama, "Adaptive networks," in *Adaptive networks*. Springer, 2009, pp. 1–8.

[13] R. Pastor-Satorras, C. Castellano, P. Van Mieghem, and A. Vespignani, "Epidemic processes in complex networks," *Reviews of modern physics*, vol. 87, no. 3, p. 925, 2015.

[14] G. Van Rossum and F. L. Drake Jr, *Python tutorial*. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands, 1995.

[15] M. Defferrard, L. Martin, R. Pena, and N. Perraudin, "Pygsp: Graph signal processing in python," Oct. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.1003158

[16] A. Hagberg, P. Swart, and D. S Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.

[17] T. E. Oliphant, *A guide to NumPy*. Trelgol Publishing USA, 2006, vol. 1.

[18] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in science & engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[19] M. Waskom, O. Botvinnik, D. O'Kane, P. Hobson, J. Ostblom, S. Lukauskas, D. C. Gemperline, T. Augspurger, Y. Halchenko, J. B. Cole, J. Warmenhoven, J. de Ruiter, C. Pye, S. Hoyer, J. Vanderplas, S. Villalba, G. Kunter, E. Quintero, P. Bachant, M. Martin, K. Meyer, A. Miles, Y. Ram, T. Brunner, T. Yarkoni, M. L. Williams, C. Evans, C. Fitzgerald, Brian, and A. Qalieh, "mwaskom/seaborn: v0.9.0 (july 2018)," Jul. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1313201

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] F. Chollet *et al.*, "Keras," https://keras.io, 2015.

[22] P. Erdös and A. Rényi, "On random graphs, i," *Publicationes Mathematicae (Debrecen)*, vol. 6, pp. 290–297, 1959.

[23] D. J. Watts and S. H. Strogatz, "Collective dynamics of small-worldnetworks," *nature*, vol. 393, no. 6684, p. 440, 1998.

[24] A. Kolla, "Cs 598: Spectral graph theory. lecture 1," 2012. [Online]. Available: https://courses.engr.illinois.edu/cs598ak/sp2012/Lecture1.pdf

[25] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *science*, vol. 286, no. 5439, pp. 509–512, 1999.

[26] A.-L. Barabási, "Scale-free networks: a decade and beyond," *science*, vol. 325, no. 5939, pp. 412–413, 2009.

[27] T. Gross and B. Blasius, "Adaptive coevolutionary networks: a review," *Journal of the Royal Society Interface*, vol. 5, no. 20, pp. 259–271, 2007.

[28] W. W. Zachary, "An information flow model for conflict and fission in small groups," *Journal of anthropological research*, vol. 33, no. 4, pp. 452–473, 1977.

[29] S. Hosseinalipour, J. Wang, H. Dai, and W. Wang, "Detection of infections using graph signal processing in heterogeneous networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[30] Y. Wang, D. Chakrabarti, C. Wang, and C. Faloutsos, "Epidemic spreading in real networks: An eigenvalue viewpoint," in *22nd International Symposium on Reliable Distributed Systems, 2003. Proceedings.* IEEE, 2003, pp. 25–34.

[31] A. V. Goltsev, S. N. Dorogovtsev, J. G. Oliveira, and J. F. Mendes, "Localization and spreading of diseases in complex networks," *Physical review letters*, vol. 109, no. 12, p. 128702, 2012.

[32] S. Fortunato, "Community detection in graphs," *Physics reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[33] S. Fortunato and D. Hric, "Community detection in networks: A user guide," *Physics reports*, vol. 659, pp. 1–44, 2016.

[34] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[35] I. Jolliffe, *Principal component analysis*. Springer, 2011.

[36] J. Shlens, "A tutorial on principal component analysis," *arXiv preprint arXiv:1404.1100*, 2014.

[37] H.-Y. Lee, "Machine learning class homework 7 - unsupervised learning," May 2019. [Online]. Available: https://www.kaggle.com/c/ml2019spring-hw7/data

[38] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *science*, vol. 313, no. 5786, pp. 504–507, 2006.