SPECTRAL ANALYSIS OF DIRECTED GRAPHS USING MATRIX PERTURBATION THEORY

by

Yuemeng Li

A dissertation submitted to the faculty of
The University of North Carolina at Charlotte
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in
Computing and Information Systems

Charlotte

2017

Approved by:

Dr. Aidong Lu

Dr. Xintao Wu

Dr. Yu Wang

Dr. Zbigniew Ras

Dr. Wei Zhao

ABSTRACT

YUEMENG LI. Spectral Analysis of Directed Graphs using Matrix Perturbation Theory.
(Under the direction of DR. AIDONG LU and DR. XINTAO WU)

The spectral space of the adjacency matrix contains important structural information of a given network (graph), where such information can be leveraged in developing a variety of algorithms in applications such as graph partition, structural hierarchy discovery, and anomaly detection. Although many prominent works have laid the foundation for studying the graph spectra, it is still challenging to analyze the spectral space properties for directed graphs due to possible complex valued decompositions. Matrix factorization techniques such as Laplacian and normalized Laplacian have been widely adopted to study the associated spectral spaces, but network structural properties may not be well preserved in those spectral spaces due to transformations.

In this dissertation work, we explore the adjacency eigenspace of directed graphs using matrix perturbation theory and examine the relationships between graph structures and the spectral projection patterns. We study how to detect dominant structures such as clusters or anomalous nodes by establishing a connection between the connectivity of nodes and the geometric relationships in the adjacency eigenspace. We leverage selected key results from perturbation theory, linear algebra and graph theory as our tools to derive theoretical results that help to elaborate observed graph spectral projection patterns. In order to validate our theoretical results, novel algorithms including spectral clustering for both signed and unsigned networks, asymmetry analysis for network dominance, and anomaly analysis for streaming network data are developed and tested on both synthetic and real datasets. The

empirical evaluation results suggest that our algorithms performs better when compared with existing state-of-the-art methods.

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to the following people that without their support, suggestions and guidance, my pursuit for this PhD degree would not be possible.

First of all, I would like to thank my advisors, Dr. Aidong Lu and Dr. Xintao Wu. They patiently guided me and provided immense support for me through my PhD studies.

I would also like to thank my committee members Dr.Yu Wang, Dr. Zbigniew W. Ras and Dr. Wei Zhao, for sharing their effort and time on my research and dissertation.

It is pleasure to study and work with labmates Dr. Xiaowei Ying, Dr. Leting Wu, Dr. Yue Wang, Boshu Ru and Kodzo Wegba in Data Privacy Lab and Visualization Center at University of North Carolina at Charlotte.

Finally, I would like to thank my family, especially my mother and wife for their love and support. This dissertation is dedicated to them.

This work is supported by the National Science Foundation under Grants No. 1047621 and 1564039.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Analysis of Online Social Networks

Social networks are constructed by the relationships between individuals, so that information or attributes associated with individuals can flow within those networks. The ties between individuals are invisible but rich information about the population involved in such social networks can be extracted and studied. According to the book "Social Network Analysis" [100], social network analysis was first introduced in sociology and then used in many other research fields. It focuses on studying and extracting the properties of the relationships between objects; differently, conventional data mining approaches focus more on the aspect of identifying the labels or properties of individuals.

Although social networks have been systematically studied since early 1930s, it was not until recently that graph theory based approaches have become principle methods for analyzing such networks [99]. Due to the fact that any given network can be translated into a graph of certain type, graph theory based approaches could be used to formulate complicated mathematical models to depict the properties of social networks. Therefore, graph theory has gradually became the dominant mathematical tool in analyzing social network related problems.

With the advance of internet technologies, online social networks such as Twitter, Facebook and LinkedIn have been flourishing and prosperous for the past two decades. Due

to the sizes of such networks, the vast amount of data generated from them drew attentions of researchers from various fields of studies such as data analytics, statistics, social science, behaviour science and others. As a result, many real world applications use the data from online social networks to perform various types of researches such as fraud detection [22, 88, 135], marketing [20, 104, 119], health [15, 113], link prediction [68, 97], visualization [124] and web mining [74, 75].

As new challenges emerge, traditional data mining approaches need to be improved and new methods need to be proposed in order to cope with them. Although many ideas and algorithms have been developed to solve relevant problems of analyzing online social networks, there are still areas that remain obscure and many open problems waiting to be solved. In addition, advanced and noval methods are still needed to push the theoretical research forward and improve the applications based on the theoretical results.

1.2 Spectral Graph Analysis and its Applications

According to [14], Leonhard Euler's paper on the Seven Bridges of Königsberg in 1736 was considered to be the first paper in the history in graph theory. Then Cauchy [37] and L'Huillier [64] studied edges and vertices and generalized their properties, so the branch of mathematics known as topology began to emerge. The term "graph" was first introduced by Sylvester in 1878, since when it becomes a standard terminology in scientific researches. The very same author also proposed the Sylvester equation problem in linear algebra regarding the linear operators related to the spectral properties of adjacency matrices, which is essential to the matrix perturbations theories [12]. Since graph theory as a mathematical tool could model very complex structures of networks, social science adopted it to build

models for solving social network related problems [43, 93].

Spectral graph theory studies the relationships of the properties of a graph with respect to the eigenpairs of its associated matrices such as adjacency matrix, Laplacian matrix or normalized Laplacian matrix [24]. Due to the keen connections of the adjacency matrices, graphs and networks, it is shown by a large quantity of research works that the network properties could be well preserved in a reduced dimension of the eigenspace of its associated matrix. As a result, such an advantage makes the spectral analysis of graphs relatively easy by reducing the exploration space significantly. Among the spectral analysis methods, Laplacian based methods have been the most widely used. Since the Laplacian matrix of the graph is a direct transformation of its associated incidence matrix, the geometric connectivity of the graph is preserved. Therefore, there are many variants of this method in recent literatures focusing on different types of graphs such as undirected, directed or signed. On the other hand, adjacency matrix based approaches have only shown their capabilities in the spectral graph analysis recently, leaving room for improvements in the future.

When compared with conventional data mining approaches, spectral graph analysis tends to identify more relationship based structural properties of the links, other than causality results of class labels of nodes. As a result, spectral methods excel at tasks such as image segmentation, community detection and visualization. However, as real world network data grow more and more complex, where the associated graphs could have mixed properties (directed weighted graphs and directed signed graphs), primitive spectral graph analysis methods could no longer handle them properly. Therefore, either improved methods or noval ideas are needed to further address those issues.

1.3 Problems Encountered

Since most of the online social network data collected form directed networks, the associated adjacency matrices are asymmetric. As the eigenvalues of the asymmetric adjacency matrix may not be real and the eigenvectors may not form orthronormal basis naturally, such networks become difficult to analyze directly. Furthermore, networks such as Epinion [63] and Slashdot Zoo [59] contain even signed information, which further complicates the analysis process. As a result, many conventional spectral methods become incapable of handling such network data. In order to solve the issues, the simplest approach is to ignore the edge direction information, treat such networks as undirected and apply the spectral methods used for undirected graph. However, the results produced may no longer be what should be contained in the original graph. Later on, methods based on Laplacian transformation and its variants began to emerge and became the predominant direction for analyzing directed graphs according to the survey [71]. Soon after, the most prominent work in spectral clustering of directed signed graphs [60] proposed the signed Laplacian approach. However, Laplacian or normalized Laplacian based approaches tend to separate negatively connected vertices rather than group positively connected vertices [23]. This causes the clustering process to produce more disputing clusters for the network than those it should have. The authors of a later work [140] proposed to use a balanced normalized Laplacian approach to solve this problem.

To summarize, the adjacency matrix of a directed network is difficult to analyze due to the existence of non-real eigenpairs; the undirected signed networks have the so called balance issue, which causes the spectral clustering process to produce unbalanced clusters;

for directed signed networks, both issues present. Those problems are what we try to address in this dissertation work.

1.4 Motivations and Contributions

The spectral space of the adjacency matrix for a given network contains key geometric information related to its underlying structure. However, such a space does not gain much attention until recently. Also, matrix perturbation theory studies the changes to the eigenpairs of a matrix when its entries change. It was shown in the works [131–133] that observed networks could be modeled as perturbations from a network with certain simple structures. Therefore, matrix perturbation theory could be used as a tool to analyze the spectral space properties of the given network. The works justified the existence a direct relationship of the geometric phenomena in the adjacency eigenspace with the underlying network structure. However, all the relevant results concern only undirected graphs, but it is foreseeable that similar properties should exist in the adjacency eigenspace of directed networks.

Although the adjacency eigenspaces of directed graphs have not been well studied before, we propose to use the very same mathematical tool to analyze such graphs and take the first step in this area. However, problems described above will make the analysis more difficult when compared with the undirected cases. After establishing the framework for directed unsigned graphs, it is further extended to directed signed graphs to completely describe adjacency spectral properties of the entire set of directed networks. We also take a different approach to study directed networks by analyzing the asymmetries of the information contained in the networks through analyzing the Singular Value Decomposition

(SVD) spectral spaces of their skew symmetric decomposition. As a result, dominance relationships could be extracted from such spectral spaces. Last but not least, we use the adjacency eigenspace properties to analyze the anomalies of dynamic graphs. With careful modeling and rigorous statistical inferences, we demonstrate how individual's anomaly measures could be influenced by the interactions between each other. To summarize, our contributions are:

1. We have explored the adjacency eigenspace of directed unsigned graphs and provided theoretical explanations for spectral phenomena in such spaces.

2. We have developed a spectral clustering based community detection algorithm for directed unsigned graphs and conducted empirical evaluations on synthetic and Twitter streaming data.

3. We have explored the adjacency eigenspace of directed signed graphs and provided theoretical explanations for spectral phenomena in such space. This step generalized the theoretical results for all directed graphs.

4. We have proposed a spectral clustering based community detection algorithm for directed signed graphs and conducted researches on synthetic and Sampson's, Slashdot Zoo, Wikisigned and Epinion datasets.

5. We have studied the asymmetry information captured in the skew symmetric decomposition and proposed a scoring method for measuring network dominance relationships. It could be used to detect the organizational hierarchy of the given relationship network. The method has been tested on synthetic data and world trade data of year

2014.

6. We have analyzed dynamic temporal network data using Vector Autoregressive (VAR) model for fraud or anomaly detection and analysis.

1.5 Thesis Organization

Given that the vast majority of Online Social Network (OSN) data today are directed or even signed directed, existing approaches could no longer be adequate to fully extract needed information from them. In this dissertation work we propose to study several topics related to directed graphs generated from OSN data.

The first step is to build a theoretical framework for directed unsigned graphs. We propose to use matrix perturbation theory to analyze the spectral coordinate changes in the perturbed Perron Frobenius simple invariant subspaces. The observed network could be perceived as departures from a K-block network structure where each cluster possesses the Perron Frobenius property according to the structure based cluster definition. We derive mathematical approximations for spectral coordinates of observed graphs using the unperturbed copy. A spectral clustering based community detection method, Augmented_ADJCluster, is produced and tested on various synthetic and real datasets.

In the second step, we extend the above framework into directed signed graphs to complete the theories for spectral analysis on directed graphs in general. We adopt the same mathematical tools to deal with the existence and absence of Perron Frobenius properties for clusters with negative entries. Since signed components may not be structurally balanced and may not have the Perron Frobenius properties, we propose a way to utilize complex spectral radii of clusters to overcome this issue. A spectral clustering algorithm

called General_ADJCluster is introduced and tested on synthetic and real datasets.

In the next step, we study the asymmetric information contained in the skew symmetric matrices, which belong to a special case of directed signed graphs. Due to the unique order preserving nature exhibited in the SVD spectral space of the skew symmetric form of any directed graph, we propose a scoring method to measure the dominance/submissiveness status of nodes, which could be used to analyze the underlying organizational hierarchies of directed networks. The method is also tested on both synthetic and real datasets.

In the last step, we apply our graph spectral analysis framework on the fraud and anomaly detection/analysis for dynamic OSN data. We build node anomaly metric time series data from the adjacency spectral features of the network snapshots. Then, we use the VAR method to model the interactions of time series data. The Granger causality test based anomaly analysis Algorithm OSN_rVAR_Granger is introduced, and several case studies based on a real and labeled streaming dataset are included to demonstrate its efficacy.

This dissertation is organized as follows:

- Chapter 2 contains preliminary background information of graph theory, linear algebra and matrix perturbation theories related to our works.

- Chapter 3 introduces how the observed directed unsigned graphs could be modeled as perturbations from networks with isolated clusters. The adjacency eigenspaces of such graphs are studied.

- Chapter 4 extends the framework in chapter 3 into directed signed graphs. Spectral behaviours under perturbations are studied in further detail to handle the more complicated issues associated with directed signed graphs.

- Chapter 5 studies the spectral properties of the SVD of the skew symmetric decomposition of relationship networks.

- Chapter 6 explores the spectral space of temporal network data using the restricted VAR model and analyzes the endogenous and exogenous influences to a node's anomaly measures using Granger causality test.

- Chapter 7 presents the conclusion and future works.

CHAPTER 2: BACKGROUND INFORMATION

2.1 Preliminaries

In this study, we focus on directed graphs without self-loops. A directed graph $G(V, E)$ can be represented as the adjacency matrix $A_{n\times n}$ with $a_{ij} > 0$ if there exists a positive edge pointing from node V_i to node V_j, $a_{ij} < 0$ if there exists a negative edge pointing from node V_i to node V_j, and $a_{ij} = 0$ otherwise. The information provided in this chapter are for general directed signed and weighted graphs. The symbols and definitions are given in Table 1.

Table 1: Symbols and Definitions

A	Adjacency matrix of a graph
P	Permutation matrix
$\widetilde{A}$	Perturbed matrix of A
$\mathcal{L}(L)$	The set of eigenvalues of L
$\Re(X)$	An invariant subspace of A spanned by a basis X
$\rho(A)$	The spectral radius of A
$(q_1, \cdots, q_n)$	An orthonormal basis of A
$(\lambda_1, \cdots, \lambda_n)$	Eigenvalues of A
$(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$	Eigenvectors of A
A^H	Conjugate transpose of A
Unitary	A is unitary if $A^{-1} = A^H$

2.1.1 Eigenspace Projection

Eigenspace projection is a method to project nodes in the spectral subspace formed by a set of eigenvectors. When chosen correctly, it will reveal the node-cluster relations of the underlying network structure. In order to perform the eigenspace projection, we need

to make sure that the eigenvectors forming the spectral space are linearly independent, otherwise the eigenspace will not be of full rank. We adopt the eigenspace projection method as in the work [132]. An illustration of the projection method is given as Equation (1). The eigenvector $\boldsymbol{x}_i$ is represented as a column vector. For undirected graphs, the eigenvectors $\boldsymbol{x}_i$ ($i = 1, \cdots, K$) corresponding to the K largest real eigenvalues contain the most topological information of the corresponding K communities of the graph in the spectral space. The K-dimensional spectral space is spanned by $(\boldsymbol{x}_1, \cdots \boldsymbol{x}_K)$. When a node u is projected in the K-dimensional subspace with $\boldsymbol{x}_i$ as the basis, the row vector $\boldsymbol{\alpha}_u = (x_{1u}, x_{2u}, \cdots, x_{Ku})$ are its coordinates in this spectral subspace. For directed graphs in this study, the chosen K eigenvectors corresponding to the Perron Frobebius simple invariant subspace will be used to perform the projections.

$$
\begin{array}{c}
\begin{array}{cccccc} \boldsymbol{x}_1 & & \boldsymbol{x}_i & & \boldsymbol{x}_K & \boldsymbol{x}_n \end{array} \\
\downarrow \\
\boldsymbol{\alpha}_u \rightarrow \left(\begin{array}{c|c|c|c}
x_{11} \cdots & x_{i1} & \cdots\ x_{K1} \cdots & x_{n1} \\
\vdots & \vdots & \vdots & \vdots \\
\hline
x_{1u} \cdots & x_{iu} & \cdots\ x_{Ku} \cdots & x_{nu} \\
\hline
\vdots & \vdots & \vdots & \vdots \\
x_{1n} \cdots & x_{in} & \cdots\ x_{Kn} \cdots & x_{nn}
\end{array}\right)
\end{array}
\tag{1}
$$

2.1.2 Perturbation Theory for Square Matrices

Spectral perturbation studies the change of the eigenpairs when the graph is perturbed. It is an excellent mathematical tool for analyzing the influence of changes to the graph

spectra. We use it to estimate the spectral projections after perturbation and verify various phenomena in the adjacency eigenspace. For a square matrix A with a perturbation E, the matrix after perturbation can be written as $\widetilde{A} = A + E$. Let λ_i be an eigenvalue of A with its eigenvector $\boldsymbol{x}_i$. For the perturbed matrix, $\tilde{\lambda}_i$ and $\tilde{\boldsymbol{x}}_i$ denote the perturbed eigenpairs.

When the matrix perturbation theory is applied to analyze the spectral properties of directed graphs, the most difficult problem is that the perturbed eigenvectors cannot be estimated using simple linear combinations of other eigenvectors, since the eigenvectors do not form orthonormal basis naturally. This problem was solved by working with spectral resolutions and using orthogonal reduction to block triangular. Therefore, the estimations for perturbed eigenvectors can be expressed by the spectral resolution of A with respect to its simple invariant subspaces. We reference the relevant definitions and theorems from [109] as follows:

Lemma 1. *Let the columns of X be linearly independent and let columns of Y span $\Re(X)^{\perp}$. Then $\Re(X)$ is an invariant subspace of A if and only if $Y^H AX = 0$. In this case $\Re(Y)$ is an invariant subspace of A^H.*

Lemma 2. *Let $\Re(X)$ be an invariant subspace of A, columns of X form an orthonormal basis for $\Re(X)$, and (X, Y) be unitary. Then the decomposition of A has the reduced form:*

$$(X, Y)^H A(X, Y) = \begin{pmatrix} L_1 & H \\ \mathbf{0} & L_2 \end{pmatrix}, \tag{2}$$

where $L_1 = X^H AX$, $L_2 = Y^H AY$, $AX = XL_1$ and $H = X^H AY$. Furthermore, eigenvalues of L_1 are the eigenvalues of A associated with $\Re(X)$. The rest eigenvalues of A are those of L_2.

Definition 1. *Let $\Re(X)$ be an invariant subspace of A, and let* (2) *be its reduced form with respect to the unitary matrix (X, Y). Denote $\mathcal{L}(L)$ as the set of the eigenvalues of L. Then $\Re(X)$ is a* **simple invariant subspace** *if $\mathcal{L}(L_1) \cap \mathcal{L}(L_2) = \emptyset$.*

With the above definition and lemmas, under the assumption of the simple invariant subspace, the approximations of perturbed eigenvectors are given by Theorem 2.7 in chapter V of [109] as follows:

Lemma 3. *Let $\widetilde{A} = A + E$, $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ be a basis of A and denote $X = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K)$ and $Y = (\boldsymbol{x}_{K+1}, \cdots, \boldsymbol{x}_n)$. Suppose that (X, Y) is unitary, and suppose that $\Re(X)$ is a* **simple invariant subspace** *of A so that it has the reduced form as Equation* (2)*. For $i \in (1, \cdots, K)$, the perturbed eigenvectors $\tilde{\boldsymbol{x}}_i$ can be approximated as:*

$$\tilde{\boldsymbol{x}}_i \approx \boldsymbol{x}_i + Y(\lambda_i I - L_2)^{-1} Y^H E \boldsymbol{x}_i, \tag{3}$$

when the following conditions hold:

1. $\delta = \inf_{\|T\|=1} \|THT\|_2 - \|X^H E X\|_2 - \|Y^H E Y\|_2 > 0$*, where $H = X^H A Y$ and $t_i \approx (\lambda_i I - L_2)^{-1} Y^H E \boldsymbol{x}_i$ for column vectors in T.*
2. $\gamma = \|Y^H E X\|_2 < \frac{1}{2}\delta$.

2.1.3 Perron-Frobenius Eigenpair

For the strongly connected component C, the following lemma shows the relationship between the connectedness and reducibility of a graph.

Lemma 4. *[123] Let A_c be the adjacency matrix representation of a component C, then C is strongly connected iff A_c is irreducible (cannot be reduced into the form of Equation* (4)*).*

$$PA_cP^{-1} = \begin{pmatrix} A_{c1} & & U \\ & \ddots & \\ \mathbf{0} & & A_{cn} \end{pmatrix}, \tag{4}$$

Furthermore, we introduce the Perron-Frobenius theorem for non-negative irreducible components.

Lemma 5. *Chapter* 8 *of [73]. Let* C *be an irreducible and non-negative* $c \times c$ *matrix corresponding to a strongly connected component. Let* $\lambda_1, \cdots, \lambda_c$ *be its (real or complex) eigenvalues. Then its spectral radius* $\rho(C)$ *is defined as:*

$$\rho(C) \stackrel{\text{def}}{=} \max_p(|\lambda_p|). \tag{5}$$

It is called the Perron-Frobenius eigenvalue of C *and the corresponding eigenvector is called the Perron-Frobenious eigenvector. The following properties hold:*

1. *The spectral radius* $\rho(C)$ *is a positive real number and it is a simple eigenvalue of* C.
2. *The only eigenvector that has all positive components is the one associated with* $\rho(C)$. *All the other eigenvectors have mixed signed components.*

Lemmas 4 and 5 simply suggest that there exists a bijective mapping from the set of communities to the set of spectral radii of all the communities. Therefore, if the network has a clear community structure, we can identify the underlying community structure by analyzing its spectral projection in the subspace spanned by Perron-Frobenius eigenvectors. This selection could essentially avoid the complex valued eigenpairs in asymmetric adjacency matrices.

2.2 Skew Symmetric Decomposition

2.2.1 Singular Value Decomposition

Let X denote an $m \times n$ matrix of real-valued data and rank r. Without loss of generality, we assume $m \geq n$. The singular value decomposition of X is

$$X = U\Sigma V^T, \tag{6}$$

where U is an $m \times n$ matrix, Σ is an $n \times n$ diagonal matrix, and V is an $n \times n$ matrix. The columns of U are called the left singular vectors and form an orthonormal basis. In other words, $UU^T = I_{m\times m}$. The columns of V are called the right singular vectors and we have $VV^T = I_{n\times n}$. The diagonal matrix $\Sigma = diag(\sigma_1, \cdots, \sigma_n)$ contains the singular values and successive singular values are monotone decreasing. Furthermore, we have $\sigma_k > 0$ for $k = 1, \cdots, r$, and $\sigma_k = 0$ for $k = r + 1, \cdots, n$.

When most or all of the variance is associated with the first few singular values, a relatively small number of spectral dimensions associated with those singular values can be used to provide an acceptably accurate depiction of the structure. According to Eckart-Young theorem, SVD can produce a low rank approximation of the given matrix with the least Frobenius norm difference under a given constrain rank [108]. In other words, $X_k = U_k\Sigma_k V_k^T$ is the closest rank-k matrix to X and X_k minimizes the sum of squares of differences of the elements of X and X_k.

2.2.2 SVD of Skew Symmetric Matrices

For a directed weighted graph where n nodes and m edges, its adjacency matrix X is an $n \times n$ asymmetric matrix. According to [41], the adjacency matrix X can have the

following decomposition: $X = Y + Z$, where $Y = \frac{1}{2}(X + X^T)$ and $Z = \frac{1}{2}(X - X^T)$. The symmetric matrix Y has each entry $y_{i,j} = \frac{1}{2}(x_{i,j} + x_{j,i})$ that captures the proximity of the pair of objects i and j. Y can be analyzed by using any method that handles symmetric adjacency matrix. Therefore, the pattern of the linkages and degree centralities of nodes can be well studied just by looking into Y. The matrix Z is a skew-symmetric matrix, i.e., its negative equals its transpose ($Z^T = -Z$). The matrix Z has each entry $z_{i,j} = \frac{1}{2}(x_{i,j} - x_{j,i})$ that captures the asymmetry or the extent to which object i dominates other j. Therefore, we can use the asymmetry information in the skew symmetric part Z to study the dominance/submissiveness relationships for a given network.

Theorem 1. *The singular decomposition of a real skew-symmetric matrix Z has the form $Z = U\Sigma JU^T$, where U is orthogonal, Σ is non-negative matrix of singular values arranged in non-increasing order along the diagonal and the singular values occur in equal pairs. Corresponding to each pair, the matrix J has a 2×2 skew-symmetric orthogonal diagonal block of the form, $J = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$.*

It is clear that the singular value decomposition of skew symmetric matrices has the same pre- and post-vectors, apart from possible changes of sign and permutation. Because of the balance between positive and negative cell values associated with skew-symmetry, the first dimension is associated with exactly the same amount of variance in the original data as is the second. Actually, every two successive singular values, σ_{2k-1} and σ_{2k}, are the same and the corresponding two successive singular vectors, $\boldsymbol{u}_{2k-1}$ and $\boldsymbol{u}_{2k}$, need to be treated as as units. The successive two-dimensional space formed by $\boldsymbol{u}_{2k-1}$ and $\boldsymbol{u}_{2k}$ is termed as the k-th bimension. The decreasing pairs of singular values impose a natural ordering on

the bimensions in decreasing order of importance.

2.2.3 Spectral Projection of Skew Symmetric Matrices

For SVD, we can generally project each node as a coordinate in the space formed by the scaled singular vectors. All rows of the singular vector matrix scaled by the squared root of singular values, $U\Sigma^{1/2}$, are plotted. In particular, the p-th row vector of $U\Sigma^{1/2}$, $(U_{p1}\sigma_1^{1/2}, \cdots, U_{pr}\sigma_r^{1/2})$, denotes the spectral coordinate of node p in the spectral space formed by the first r singular vectors.

For a skew symmetric matrix, each two successive singular vectors share the same singular value and need to be treated as a unit for projection. According to the canonical analysis of asymmetry [40], we need to project nodes to each bimension. Hence, the projection of node p in the first bimension is $(U_{p,1}\sigma_1^{1/2}, U_{p,2}\sigma_2^{1/2})$, its projection in the second bimension is $(U_{p,3}\sigma_3^{1/2}, U_{p,4}\sigma_4^{1/2})$, and so on.

The spectral coordinates of two nodes then capture the asymmetry between them. However, the conventional Euclidean distance is no longer appropriate. As suggested by the work [41], the amount of asymmetry between objects p and q should be represented by the area of the triangle enclosing point p, point q, and the origin o. As a matter of fact, we use the area formed with the objects and the origin to measure the difference of two objects in a 2-dimensional planar space. This coincides with the Euclidean distance in the Euclidean space. Note that the work [92] adopted the same idea to quantify the differences between the projections of two nodes in the planar space. However, they only used skew symmetry to analyze the asymmetry of geological successions, where the difference of objects was treated as non-directional. In our work, we will address the direction issues associated with

the asymmetry problems.

Another key property of SVD is that the ordering relationships of each pair of individuals are preserved in the spectral projection space. The work [33] demonstrated that SVD can be used to uncover organizational hierarchies from the skew symmetric part Z of a total ordering matrix constructed from the original matrix. Therefore, the spectral space produced by SVD could capture the dominance relationships for a given network that have total or partial ordering relationships such as orders, votes, prestige ranks. The mathematic properties of skew symmetry have been studied in the works [17, 31, 118]. The works [96, 136] in the field of image processing also studied its geometric properties. However, those works did not investigate its spectral properties.

CHAPTER 3: SPECTRAL PROPERTIES OF DIRECTED UNSIGNED GRAPHS

The eigenspace of the adjacency matrix of a graph possesses important information about the network structure. However, analyzing the spectral space properties for directed unsigned graphs (DUGs) is challenging due to complex valued decompositions. In this chapter, we explore the adjacency eigenspaces of directed unsigned graphs. With the aid of the graph perturbation theory, we emphasize on deriving rigorous mathematical results to explain several phenomena related to the eigenspace projection patterns that are unique for DUGs. Furthermore, we relax the community structure assumption and generalize the theories to the perturbed Perron-Frobenius simple invariant subspace so that the theories can adapt to a much broader range of network structural types. We also develop a graph partitioning algorithm and test it on both synthetic and real data sets to demonstrate its potential.

3.1 Analysis of Spectral Spaces of Directed Unsigned Graphs

For many non-random graphs generated from online social networks, economic networks or biological networks, there usually exist clusters (communities) formed by individuals. Identifying such structures can help us better understand properties of those networks. Researchers have developed approaches and algorithms to deal with the clustering in DUGs [9, 25, 53, 62, 72, 76, 79, 122, 141, 142] because relationships in many networks are asymmetric. Refer to [71] for a recent survey. Roughly speaking, they can be classified into

two categories. In the first category, DUGs are converted into an undirected ones, either unipartite or bipartite, where edge direction is preserved, e.g., via edge weights of the produced unipartite graph [95] or edges in the produced bipartite graph [143]. Clustering algorithms for undirected weighted graphs are then applied. Methods in the second category are mainly based on the idea of extending clustering objective functions and methodologies to DUGs. In those approaches, the graph clustering is expressed as an optimization problem and the desired clustering properties are captured in the modified objective criterion. For example, researchers developed the directed versions of modularity [53, 62, 83], the objective function of weighed cuts in DUGs [72], and the spectral graph clustering based on the Laplacian matrix of the DUGs [25, 142]. However, it is unclear to what extent the information about the directionality of the edges is retained by these approaches.

In this chapter, we study whether we can directly analyze the spectral properties of the adjacency matrix of the underlying DUGs instead of transforming the DUGs to undirected or developing the directed versions of the objective criterion used in graph clustering. When the concern is with DUGs, one main difficulty for spectral clustering is to deal with the complex values for eigenpairs associated with the asymmetric adjacency matrix. The problem of how to select a set of eigenvectors to produce a meaningful partition result becomes very complicated. Furthermore, the other major difficulty associated with analyzing the spectral spaces of asymmetric adjacency matrices is that the eigenvectors do not form an orthonormal basis naturally. This complicates the process of analyzing the behaviors of nodes in the spectral space. Although the authors in the work [132] demonstrated that the geometric properties of nodes with respect to the communities in the spectral spaces can be described perfectly using the matrix perturbation theory for undirected graphs, the situation

would be much more complicated for DUGs.

We conduct theoretical analysis to address the above difficulties by leveraging the spectral graph perturbation theory. The spectral graph perturbation focuses on analyzing the changes in the spectral space of a graph after new edges are added or deleted. We provide a theoretical analysis of the properties of the eigenspace for DUGs and develop a method to circumvent the issue of complex eigenpairs. Our analysis utilizes the connectedness property of the components of a network to screen out irrelevant eigenpairs and thus eliminating the need for dealing with complex eigenpairs. We demonstrate how to derive the approximations of the eigenvectors by leveraging the constructed orthonormal basis when treating the graph as a perturbation from a block matrix. Furthermore, the derived theories are generalized to perturbed Perron-Frobenius simple invariant subspace. The significance of such a spectral subspace is that it is a real subspace with some unique properties that contains all the spectral clustering information of a graph.

Spectral clustering based partition algorithms require one to find a correct set of eigenvectors for spectral projection. This leads to the search for a set of eigenvectors that can capture the structural information in the spectral domain. Objective optimization approaches such as modularity maximization [81], modified versions for DUGs in [62, 83], and some variants [72, 103] were proven to be effective in partitioning graphs with clear community structures according to the density based criterion, but those approaches could not fully suffice as the objective for studying structural properties of DUGs according to the pattern based criterion [71, 98]. The pattern based criterion should take priority in defining community structures. Therefore, we first make the assumption that a community should be a strongly connected component. In the later sections, we relax this assumption to a

group of nodes with a strongly connected core component and show that this relaxation is valid both by observations and in theory. Based on our theoretical analysis, we develop a novel graph partitioning algorithm that could deal with DUGs without clear community structures. There is one interesting observation that the overlapping of communities can be detected by adjusting the objective function in the algorithm if the eigenvectors are selected according to the proposed method. The algorithm is validated with synthetic data and a streamed dataset from Twitter.

To summarize, in this chapter, we provide a thorough analysis of the properties of the spectral space for DUGs, propose a method to deal with the issue of complex eigenpairs by selecting a spectral subspace spanned by a unique set of real eigenvectors, use matrix perturbation theory to rigorously prove that the perturbed Perron-Frobenius invariant subspace can indeed capture the structural properties of any given DUGs in the spectral domain, develop an algorithm to partition DUGs without transforming the adjacency matrices or modifying the objective functions, and test the algorithm on various synthetic and real data sets to demonstrate its potential.

3.2 Modeling Observed Graphs As Perturbations

We assume that the observed graph $\widetilde{A}$ of a network has K communities namely $C_1, \cdots,$ C_K. According to the pattern based criterion, we make the assumption that each community C_i in a directed graph should be a relatively dense strongly connected component [1]. This assumption makes an easy starting point to study directed graphs and will be relaxed in later sections.

[1] A component is strongly connected if there exists a path for any nodes V_i to V_j of the component.

Formally, any observed directed graph $\widetilde{A}$ containing multiple communities with the above defined community structure can be regarded as the perturbation form the diagonal block form of A as:

$$\widetilde{A} = A + E = \begin{pmatrix} A_1 & & 0 \\ & \ddots & \\ \mathbf{0} & & A_K \end{pmatrix} + E, \tag{7}$$

where A contains the communities A_is and E contains the edges connecting A_is.

3.2.1 Disconnected Communities in Directed Unsigned Networks

Given an unsigned network with K disconnected communities $C_1, \cdots, C_K$, and its adjacency matrix is expressed in the form of Equation (7). We have the following results:

Lemma 6. *For an adjacency matrix A of a graph with K disconnected communities in the form of Equation* (7)*. For $i = 1, \cdots, K$, the following results hold:*

1. *The K Perron-Frobenius eigenvalues λ_{Ci}s corresponding to communities C_is are real, positive, simple eigenvalues, and are also the eigenvalues of A.*

2. *Furthermore, let $\boldsymbol{x}_{C_i}$ be the Perron-Frobenius eigenvectors of communities, the eigenvectors $\mathbf{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K)$ of A corresponding to λ_{Ci}s are the only eigenvectors whose non-zero components are all positive, all the entries of $\mathbf{x}$ are real valued and*

have the following form:

$$(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_K) = \begin{pmatrix} \boldsymbol{x}_{C_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{x}_{C_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{x}_{C_K} \end{pmatrix}$$

.

3. *There is only one location of the row vector $\boldsymbol{\alpha_u}$ that has a non-zero value with the form:*

$$\boldsymbol{\alpha_u} = (0, \cdots, 0, \boldsymbol{x}_{iu}, 0, \cdots, 0). \tag{8}$$

The location of x_{iu} indicates the community which node u belongs to and the value of x_{iu} denotes the influence of node u to that community.

Since the matrix A is of diagonal block form, the eigenvectors of A will be of the same form corresponding to each block and the eigenvalues of A will be the union of those of A_is. The results follow from applying Lemma 4 and Lemma 5. If we perform the spectral projection as Equation (1) using this set of eigenvectors, nodes from different communities will form orthogonal lines (The cosine value of any two nodes from different communities will be 0). In the next part, we will demonstrate how the node spectral projections behave when the Perron-Frobenius spectral subspace is perturbed.

3.2.2 Observations in Perturbed Spectral Space for Directed Unsigned Graphs

To illustrate various phenomena in the perturbed Perron-Frobenius spectral subspace for unsigned graphs, we generated a toy graph with 25 nodes containing 3 communities: $C1$, $C2$ and $C3$. $C1$ contains nodes labeled 1 to 8, 14 and 15. $C2$ is an isolated community

that contains nodes labeled 16 to 25. $C3$ contains nodes 10, 12 and 13 with nodes 9 and 11 as leaf nodes. The graph is shown in Figure 1, where dashed lines represent the added cross-community edges.

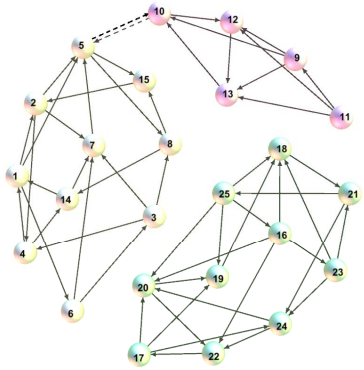
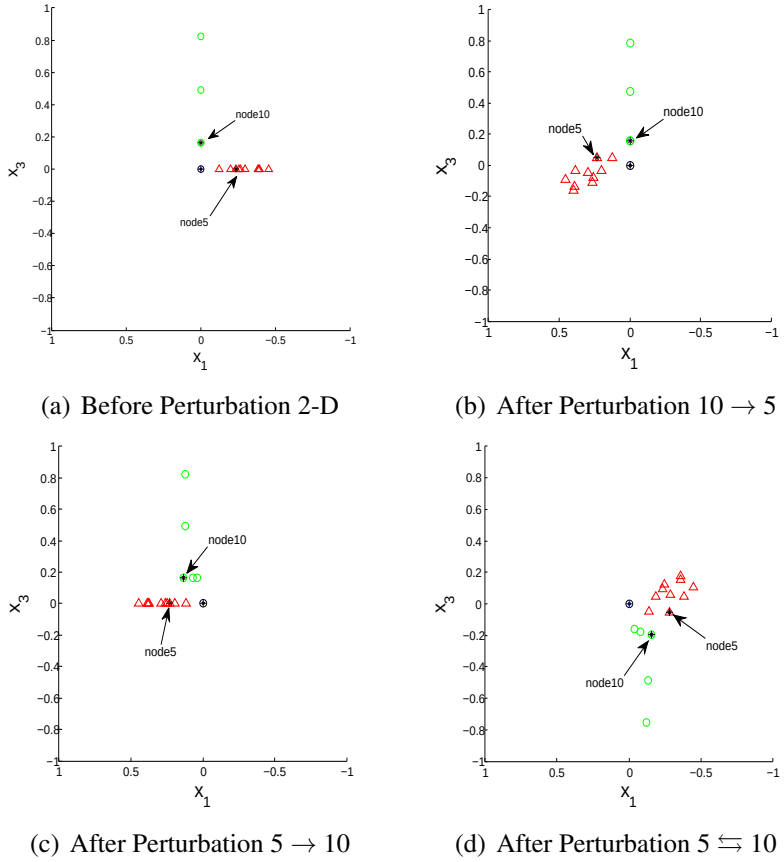


Figure 1: Perturbed Graph

In order to thoroughly examine the spectral properties of nodes, we first added an edge from node 10 in $C3$ to node 5 in $C1$, then replaced it with a reverse edge, and finally added an undirected edge. The changes to the original isolated components are treated as as perturbations.

Figure 2 demonstrates the cross sectional spectral projections for communities $C1$ and $C3$ before and after perturbations. When we look into this Perron-Frobenius spectral subspace, before the perturbation, the nodes form straight lines according to the communities they belong to and the coordinates of nodes from different communities form orthogonal lines, as shown in Figure 2(a). After we added one edge from node 10 in $C3$ to node 5 in $C1$, it can be observed clearly from Figure 2(b) that the spectral coordinates of the nodes in the two communities connected by the edge have changed. However, different from those in undirected graphs as demonstrated in [132], the nodes in the community $C3$ with an out-going edge are leaning towards the community ($C1$) that the edge is pointing to. On



(a) Before Perturbation 2-D

(b) After Perturbation 10 $\rightarrow$ 5

(c) After Perturbation 5 $\rightarrow$ 10

(d) After Perturbation 5 $\leftrightarrows$ 10

Figure 2: Spectral Coordinates of Nodes under Perturbation

the other hand, the spectral coordinates of nodes in the community ($C1$) with an incoming edge still lie on their original axis but their values on that axis have changed. This phenomena can also be observed in Figure 2(c) when we added a directed edge from node 5 in $C1$ to node 10 in $C3$. When an undirected edge (5 $\leftrightarrows$ 10) is added, Figure 2(d) resembles the combination of Figure 2(b) and Figure 2(c).

As observed in Figure 2, the signs of components of some eigenvectors have flipped. Such phenomena can be caused partly by the eigen-decomposition method and partly by the structure of the given graph. However, in this example the first cause is the reason and it will not affect the partition results in general. Detailed analyses of the second cause will be presented in Section 3.3.5. In the example given, the adjacency eigenspace revealed the

direction of the flow of information (from unperturbed to perturbed). Therefore, analyzing the adjacency eigenspace of a directed graph can reveal more about the structure changes due to the addition or deletion of directed edges such as following a Tweet, voting for a person or malicious attacks on a community.

Table 2: Eigenvectors Before and After Perturbation

	no edge			edge $5 \to 10$		edge $10 \to 5$		edges $5 \leftrightarrows 10$	
node	x_1	x_2	x_3	$\tilde{x}_1$	$\tilde{x}_3$	$\tilde{x}_1$	$\tilde{x}_3$	$\tilde{x}_1$	$\tilde{x}_3$
5	-0.2346	0.0000	0.0000	0.2281	0.0000	-0.2346	-0.0499	-0.2835	0.0513
6	-0.2666	0.0000	0.0000	0.2592	0.0000	-0.2666	0.1081	-0.2439	-0.1222
7	-0.1210	0.0000	0.0000	0.1177	0.0000	-0.1210	-0.0499	-0.1390	0.0468
8	-0.2563	0.0000	0.0000	0.2491	0.0000	-0.2563	0.0748	-0.2328	-0.0930
9	0.0000	0.0000	0.4932	0.1254	0.4932	0.0000	-0.4738	-0.1336	0.4857
10	0.0000	0.0000	0.1644	0.1364	0.1644	0.0000	-0.1579	-0.1575	0.1941
11	0.0000	0.0000	0.8220	0.1198	0.8220	0.0000	-0.7896	-0.1219	0.7518

Some of the subtle changes of spectral coordinates mentioned above may be difficult to observe from the figures directly, so we provide the spectral coordinates corresponding to the related nodes before and after the perturbations in Table 2, where irrelevant nodes are omitted to save space. In the next step, we will present our theoretical studies of the spectral analysis for DUGs.

3.3 Perturbed Eigenspace

As discussed in Section 2.1.2, the set of eigenvectors from DUGs do not form orthonormal basis naturally. The perturbation theory, introduced in Lemma 3 requires the simple invariant subspace to produce a similarity reduction of the asymmetric adjacency matrix. Hence, in order to give explicit approximations explaining the spectral projection patterns observed, we need to find a unitary orthonormal basis that satisfies the conditions in Lemma 1 and Definition 1 to achieve the orthonormal reduction for a given asymmetric matrix. It is important to emphasize that such process is not needed for undirected graphs because

the eigenvectors form orhtonormal basis naturally.

3.3.1 Orhtonormal Basis Construction

The following proposition sets up such a basis by using the Gram-Schmidt process.

Proposition 1. *Let* $\Lambda = \mathrm{diag}(\lambda_1, \cdots, \lambda_n)$ *be the eigenvalues for* $\widetilde{A}$*, and* $\mathbf{x} = (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n)$ *be the eigenvectors. Assume that all the eigenvectors are linearly independent, and, without loss of generality, let* $\sigma = (\boldsymbol{q}_1, \cdots, \boldsymbol{q}_n)$ *be an orthonormal basis formed by Gram-Schmidt process. Suppose that there exist* K *eigenvectors* $\boldsymbol{x}_i = \boldsymbol{q}_i$ *for* $i \in (1, \cdots, n)$ *that are part of this orthonormal basis and relabel their indices as* $(\boldsymbol{q}_1, \cdots, \boldsymbol{q}_K)$ *along with the corresponding eigenvalues. Denote* $X = (\boldsymbol{q}_1, \cdots, \boldsymbol{q}_K)$ *and* Q *as the rest of the orthonrmal basis. If* $\lambda_1, \cdots, \lambda_K$ *are simple, then the following results hold:*

1. $(X, Q)^H = (X, Q)^{-1}$ *is unitary.* $Q^H A X = 0$*, thus* $\Re(X)$ *is a simple invariant subspace of* A*.*

2. A *can be reduced to a block triangular form:*

$$(X, Q)^H A (X, Q) = \begin{pmatrix} L_1 & H \\ \mathbf{0} & L_2 \end{pmatrix}, \tag{9}$$

   where $L_1 = X^H A X$*,* $L_2 = Q^H A Q$ *is upper triangular,* $AX = X L_1$ *and* $H = X^H A Q$*. The eigenvalues of* L_1 *are the eigenvalues of* A *associated with* $\Re(X)$*. The rest eigenvalues of* A *are those of* L_2*.*

Proof. For (1), since (X, Q) is the orthonormal basis formed by Gram-Schmidt process, so $(X, Q)^H = (\boldsymbol{x}_i, Q)^{-1}$ is unitary.

For (2), Since X are eigenvectors of A, then $Q^H A\boldsymbol{x}_i = Q^H \lambda_i \boldsymbol{x}_i = \lambda_i \mathbf{0} = \mathbf{0}$ for each $i \in (1, \cdots, K)$. Since λ_is are simple eigenvalues, then $\mathcal{L}(L_1) \cap \mathcal{L}(L_2) = \emptyset$. By Lemma 1 and Definition 1, $\Re(X)$ is a simple invariant subspace of A. By Lemma 2, A can be reduced to a block triangular form in Equation (2). Hence, Equation (9) holds. By the method how (X, Q) was formed, due to the mechanism of Gram-Schmidt process, we have $(X, Q)R = \mathbf{x}$ where R and R^{-1} is strick upper triangular. Hence, $(X, Q)^H A(X, Q) = R\Lambda R^{-1}$ is the result of the orthonormal reduction and is upper triangular, then so is L_2. The last part is from Lemma 2. □

By item 2 of Proposition 1, the orthornormal reduction results in an upper triangular matrix. In the symmetric case, the result is a diagonal matrix containing only eigenvectors, since the eigenvectors diagonalize the matrix. In the next section, we will give the approximations for the perturbed Perron-Frobenius eigenvectors corresponding to the K communities that span the K dimensional subspace. The approximations will be used to explain several phenomena in this particular subspace.

3.3.2 Approximation

When we treat the observed graph as the perturbed graph from Equation (7), we are able to 1) use Lemma 5, Lemma 6 and Proposition 1 to show the Perron-Frobenius eigenvectors $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K)$ form a **simple invariant subspace**; and 2) use the perturbation theory shown in Lemma 3 to derive the approximation of the perturbed Perron-Frobenius subspace.

Theorem 2. *Let the observed graph be $\widetilde{A} = A + E$ with K communities and the perturbation E denotes the edges connecting communities $C_1, \cdots, C_K$. Assume that E satisfies the conditions in Lemma 3. Let $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K)$ be the relabeled Perron-Frobenius eigenvectors*

of A for all communities, and Q be the rest of the orthronormal basis constructed using Proposition 1. Then $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K)$ is a simple invariant subspace of A, and the perturbed Perron-Frobenius spectral space for $\widetilde{A}$ can be approximated as:

$$(\tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_K) \approx (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K) + \nabla E(\frac{\boldsymbol{x}_1}{\lambda_1}, \cdots, \frac{\boldsymbol{x}_K}{\lambda_K}). \tag{10}$$

where $\nabla = Q(I - \frac{L_2}{\lambda_i})^{-1}Q^H$.

Proof. Let the Perron-Frobinues eigenvectors be $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K)$. Noticing that this set of eigenvectors are orthogonal before the perturbation occurs, so if we construct an unitary orthonormal basis as in Proposition 1, those eigenvectors are part of the unitary orthonormal basis and $\boldsymbol{q}_i = \boldsymbol{x}_i$ for $i \in (1, \cdots, K)$, where the indexes are relabeled to correspond to each community.

By Lemma 5, all eigenvalues corresponding to such a spectral subspace are simple. By Definition 1 and Proposition 1, $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K)$ is a simple invariant subspace of A.

By Proposition 1, Q is the rest of the orthonormal basis. Therefore, by applying Lemma 3, each of the perturbed Perron-Frobenius eigenvectors can be approximated as:

$$\tilde{\boldsymbol{x}}_i \approx \boldsymbol{x}_i + \nabla E \frac{\boldsymbol{x}_i}{\lambda_i},$$

where $\nabla = Q(I - \frac{L_2}{\lambda_i})^{-1}Q^H$.

Putting K columns together, the approximation of such a perturbed spectral space are:

$$(\tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_K) \approx (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K) + \nabla E(\frac{\boldsymbol{x}_1}{\lambda_1}, \cdots, \frac{\boldsymbol{x}_K}{\lambda_K}).$$

□

When spectral projection is performed on the subspace spanned by the eigenvectors

corresponding to the K Perron-Frobenius eigenvalues, we can use Theorem 2 to derive the approximation of spectral coordinate of $\boldsymbol{\alpha}_u$ using the following simplified result that only takes into account of the influences of neighboring nodes from other communities. Since the edge direction indicates the flow of information, we define the outer community neighbours of a node $u \in C_i$ to be any node $v \notin C_i$ that has an edge pointing to u.

Theorem 3. *For node $u \in C_i$, let Γ_u^j denote its set of neighbors in C_j for $j \in (1, \cdots, K)$. The simplified spectral coordinates $\boldsymbol{\alpha}_u$ can be approximated as:*

$$\boldsymbol{\alpha}_u \approx x_{iu} I_i + \left(\sum_{j=1}^{n} \nabla_{uj} \sum_{v \in \Gamma_u^1} \frac{e_{jv} x_{1v}}{\lambda_1}, \cdots, \sum_{j=1}^{n} \nabla_{uj} \sum_{v \in \Gamma_u^K} \frac{e_{jv} x_{Kv}}{\lambda_K} \right), \tag{11}$$

where I_i is the i-th row of a K-by-K identity matrix, e_{jv} is the (j, v) entry of E and ∇ is defined in Theorem 2.

Proof. By Theorem 2, the perturbed spectral space have the form:

$$(\tilde{\boldsymbol{x}}_1, \cdots, \tilde{\boldsymbol{x}}_K) \approx (\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K) + \nabla E(\frac{\boldsymbol{x}_1}{\lambda_1}, \cdots, \frac{\boldsymbol{x}_K}{\lambda_K}).$$

Then, by Lemma 6,and Equation (8), the spectral coordinate of node u can be simplified by only considering the influences by neighbours from other communities:

$$\begin{aligned}
\boldsymbol{\alpha}_u &\approx x_{iu}(0, \cdots, 1_i, \cdots, 0) \\
&\quad + \left(\sum_{j=1}^{n} \nabla_{uj} \sum_{v \in C_1} \frac{e_{jv} x_{1v}}{\lambda_1}, \cdots, \sum_{j=1}^{n} \nabla_{uj} \sum_{v \in C_K} \frac{e_{jv} x_{Kv}}{\lambda_K} \right) \\
&\approx x_{iu} I_i \\
&\quad + \left(\sum_{j=1}^{n} \nabla_{uj} \sum_{v \in \Gamma_u^1} \frac{e_{jv} x_{1v}}{\lambda_1}, \cdots, \sum_{j=1}^{n} \nabla_{uj} \sum_{v \in \Gamma_u^K} \frac{e_{jv} x_{Kv}}{\lambda_K} \right),
\end{aligned}$$

where I_i is the i-th row of a K-by-K identity matrix. □

The entry $\sum_{j=1}^{n} \nabla_{uj} \sum_{v \in \Gamma_u^i} \frac{e_{jv} x_{iv}}{\lambda_i}$ in the i-th column position of the spectral coordinate in Equation (11) is responsible for determining the influence of the perturbation to the current community members. For general perturbation, the perturbation could occur inside the community or even onto the node itself. Therefore, for the spectral coordinates of node u, this term will be 0 only when the perturbation does not appear on the column positions of E corresponding to the community which the node u belongs to. On the other hand, if perturbations occur inside the column positions of E corresponding to the community where the node u is, the values of $\boldsymbol{\alpha}_{iu} (\forall u \in C_i)$ will be altered. This phenomenon is reasonable, since all members in a community are strongly connected. Hence, the perturbation influence affects the entire community.

3.3.3 Inference

With Theorem 3, we can make the following analysis and explain some of the phenomena observed from Figure 2 and Table 2. Before the perturbation, when the adjacency matrix A is of the diagonal block form, the second part of right hand side of Equation (11) will be 0, so nodes from the community C_i will lie on line I_i. Since $I_i \cdot I_m = 0$ for $i \neq m$, the nodes from different communities lie on different orthogonal lines. After the matrix is perturbed, suppose that the perturbation happens on the C_m region of the vth column of E, then $E\boldsymbol{x}_i = 0$, since the C_m region of $\boldsymbol{x}_i = 0$ by Equation (8). Then the coordinates of all the nodes in C_i with respect to the two-dimensional subspace become $(\boldsymbol{x}_{iu}, \sum_{j=1}^{n} \nabla_{uj} \sum_{v \in \Gamma_u^m} \frac{e_{jv} x_{mv}}{\lambda_m})$ for $u \in C_i$. Likewise, the coordinates of the nodes in C_m are: $(0, \boldsymbol{x}_{mw} + \sum_{j=1}^{n} \nabla_{wj} \sum_{v \in \Gamma_u^m} \frac{e_{jv} x_{mv}}{\lambda_m})$ for $w \in C_m$. The dot products of any two rows are not 0, so the projections of nodes do not form strict orthogonal lines. Due to the sum of

the product of scalar ∇_{ij} and the remaining terms, the spectral projections of all the nodes u of the same community C_i will deviate from the original line at different rates depending on the values in ∇.

The following is an example with a graph of two communities to illustrate the above proposition. Suppose nodes u and v are from community C_1 and C_2 respectively, the perturbation matrix E adds an edge from u to v as $u \rightarrow v$. Then the spectral coordinates for nodes u and v in the two dimensional space would be:

$$\begin{pmatrix} \boldsymbol{x}_{1u} & \nabla_{u1}\frac{e_{uv}}{\lambda_2}x_{2v} \\ \mathbf{0} & \boldsymbol{x}_{2v} + \nabla_{v1}\frac{e_{uv}}{\lambda_2}x_{2v} \end{pmatrix}. \tag{12}$$

This result coincides precisely with the observed pattern in the case in which and edge is added from node 10 to node 5 in Figure 2 and the data shown in Table 2. Therefore, we have confirmed using matrix perturbation theory that the spectral space spanned by perturbed Perron-Frobenius eigenvectors in observed graphs can indeed capture the underlying community structures.

3.3.4 Core of a Community

At the beginning of this chapter, we made the assumption that communities are strongly connected components. However, this requirement is unrealistic in real-world applications because nodes within one community may not be strongly connected. In our illustrative example as shown in Figure 2 and Table 2, nodes 9 and 11 were not members of any strongly connected component, but they could still be clustered into the strongly connected component which they point directly to with edges. Our theoretical result based on the matrix perturbation can be extended to a general case.

Theorem 4. *Suppose that a community has a large strongly connected core and a small portion of leaf nodes which all have edges pointing to the members in the core. Let the leaf edges be a perturbation matrix E and treat the core as A. If the norm of E satisfies the conditions in Lemma 3, the leaf nodes will have values in the corresponding locations of the perturbed Perron-Frobenius eigenvector of A. Hence all the nodes can be clustered according to their correlations with communities in the perturbed Perron-Frobenius spectral subspace.*

Proof. Since the norm of E satisfies the conditions in Lemma 3. All the nodes will have values in the corresponding positions of the perturbed Perron-Frobenius eigenvectors. Therefore, nodes can be clustered according to the their corresponding eigenvectors in the perturbed Perron-Frobenius spectral subspace. Apply Theorem 3 and consider Equation (12). The spectral coordinates for leaf node u pointing to node v in community 2 are:

$$\begin{pmatrix} \mathbf{0} & \nabla_{uu}\frac{e_{uv}}{\lambda_2}x_{2v} \\ \mathbf{0} & \boldsymbol{x}_{2v} + \nabla_{vu}\frac{e_{uv}}{\lambda_2}x_{2v} \end{pmatrix}. \tag{13}$$

we get the coordinates on the community $j \in (1, \cdots, K)$ for any such leaf nodes u as $\nabla_{uu}\frac{e_{uv}}{\lambda_j}x_{jv}$. If one leaf node points to multiple communities,it will be clustered to the one with the most edge weight. This is equivalent as the eigenvector centrality. □

Based on this theorem, we can replace the original assumption that all communities are strongly connected components with a weaker one: if all communities have strongly connected cores, all nodes can be assigned to communities based on the geometric relationships in the spectral space spanned by the perturbed Perron-Frobenius eigenvectors.

3.3.5 Perturbation Influences to Eigenvectors

In general, when $\|E\|_2$ is small, nodes from the same community form a cluster towards a fitted line by using the spectral projection method according to our theories above. Therefore, nodes can be clustered in the spectral space based on their spectral coordinates when $\|E\|$ is small, but the clustering results would degrade as $\|E\|$ grows. When a larger community points an edge to a smaller community, the Perron-Frobenius eigenvector corresponding to the smaller community will be altered tremendously, since the inflow perturbation from the eigenvector corresponding to the larger spectral radius completely dominates the perturbed eigenvectors of the smaller community.

Recall the fourth case in Figure 2, two communities were joined by an undirected edge to become one single strongly connected component. In this case, the perturbed eigenvectors corresponding to the larger spectral radius will become the Perron-Frobenius eigenvector of the newly formed strongly connected component, thus it will be strictly positive. However, the Perron-Frobenius eigenvector corresponding to the smaller community will be mixed signed. Our following corollary explains mathematically why some observed sign changes in eigenvectors.

Corollary 1. *The sign changes in perturbed eigenvectors $\tilde{\boldsymbol{x}}_i$ are influenced by the corresponding spectral radius λ_j of the neighbouring community C_j. There are two cases:*

1. *The values in $\tilde{\boldsymbol{x}}_i$ remain same signed when $\lambda_i \gg \lambda_j$.*

2. *The values in $\tilde{\boldsymbol{x}}_i$ will be mixed signed when $\lambda_i \leq \lambda_j$.*

Proof. A simplified mathematical representation for this phenomena can be explained by

using Equation (12).The perturbed eigenvector of it can be approximated as:

$$\tilde{\boldsymbol{x}}_i \approx \boldsymbol{x}_i + Q(I - \frac{L_2}{\lambda_i})^{-1} Q^H E \frac{\boldsymbol{x}_i}{\lambda_i}.$$

Since the diagonal elements of L_2 corresponds to spectral radii of neighbouring communities according to Proposition 1, then the term $(I - \frac{L_2}{\lambda_i})^{-1}$ would be determined by the incoming influences from other communities.

In the first case, when $\lambda_i \gg \lambda_j$, the perturbation will have minimal influence, thus the values in $\tilde{\boldsymbol{x}}_i$ will not be perturbed too much and will keep their original signs.

In the later case, the perturbation results will be completely dominated by the incoming influence, so $(I - \frac{L_2}{\lambda_i})$ will contain all negative values on its diagonal. Therefore, most of the locations in $\tilde{\boldsymbol{x}}_i$ corresponding to those of nodes in C_j will have different signed values. This caused the values of $\tilde{\boldsymbol{x}}_i$ to be mixed signed. □

3.4 Algorithm

According to Theorem 3, we have $\boldsymbol{q}_i = \boldsymbol{x}_i$ for $i \in (1, \cdots, K)$, so $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K)$ are part of the orthornormal basis. Then by Lemma 5 and Definition 1, $(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_K)$ is a simple invariant subspace. Hence, all the results derived for observed graphs can be generalized into the perturbed spectral space from the simple invariant subspace spanned by the Perron-Frobenius eigenvectors. By combining the results with Perron-Frobenius Theorem, this particular simple invariant subspace has many unique properties before perturbations: it is real valued, values in each column vector are same signed with small or no incoming perturbations, and its dimension equals the number of the communities. Furthermore, it contains the some of the most important topological information of a given graph, since

the partition in this spectral space maximizes the modularity score under same assumptions in [81, 83] and their variants.

With all the observations and theoretical results, a spectral clustering algorithm follows immediately, as shown in Algorithm 1. Our algorithm includes the following major steps: diagonalization of the adjacency matrix; normalization of the eigenvectors; selecting the initial set of eigenvectors with same signed components whose corresponding eigenvalues are real valued, positive and amongst the largest real positive eigenvalues of the adjacency matrix; projection of the nodes onto a unit sphere; clustering the nodes according to their location on the unit sphere using K-means; screen all the potential eigenpairs based on the modularity to find meaningful partitions.

As discussed previously, there are several factors that can affect the signs of the the components of the perturbed eigenvectors. Therefore, the initial set of eigenpairs may not include all the perturbed Perron-Frobenius eigenvectors. As a result, we need to search through all the real eigenvectors to select the ones that could increase the modularity. This process will cross validate all the newly added eigenvectors with the selected set of perturbed Perron-Frobenius ones. As a result, it would reduce the workload while avoiding producing a partition that deviates from the true structure by selecting non-Perron-Frobenius eigenvectors in the beginning. Since the communities in DUGs are not defined by the density based criterion, maximizing modularity could no longer suffice as the objective. Ideally, we could use a combination of objective functions to determine the communities, but due to limited space, we will only test modularity with a tuning factor α. The rationale behind this approach is: although adding some eigenvector to partition the graph reduces the overall modularity by an insignificant amount, this partition could still be

meaningful. As it turns out in our empirical evaluation, this approach can detect overlaps of communities if used properly.

Algorithm 1 *Augmented-AdjCluster*: Simple Invariant Subspace Based Spectral Clustering for DUGs

Input: A, τ, α
Output: clustering result CL

Compute eigenvectors of A corresponding to the largest τ real eigenvalues, and denote this set as E and their eigenvalues Λ;
Normalize the eigenvectors $\bar{\boldsymbol{\alpha}}_u = \frac{\boldsymbol{\alpha}_u}{\|\boldsymbol{\alpha}_u\|}$;
$C \leftarrow$ eigenvectors from E with same signed components;
$S \leftarrow Cardinality(C)$;
$M \leftarrow 0$;
for each $c \in \emptyset \cup E \setminus C$ **do**
    Apply k-means algorithm on $C \cup c$ to get clustering result R;
    Compute the Modularity scores M_{temp};
    if $M_{temp} \geq \alpha M$ ($\alpha \in [0, 1]$ adjusts the objective function) **then**
        $S \leftarrow S + 1$;
        $C \leftarrow C \cup c$;
        $CL \leftarrow R$;
        $M \leftarrow M_{temp}$;
    end if
end for
Return S and clustering result CL;

3.5 Empirical Evaluation

In this evaluation, we mainly compare our algorithm *Aug_Adj* with several representative spectral clustering methods. In particular, we compare with the random walk based Normalized cut (Ncut) [142], the random walk based Laplacian method (Lap) [72], the adjacency matrix based method using symmetrization (AdjCl) [132], and the SVD based method which works on the eigenspace associated with $A^H A$ and AA^H (SpokeEn) [91] on synthetic graphs under various conditions. Note that both Ncut and Lap are spectral clustering methods for DUGs and the transition matrices used there are based on the classic PageRank method. In our evaluation, we set the default damping factor 0.85 in the

PageRank when calculating the transition matrices used in Ncut and Lap.

3.5.1 Synthetic Data

The synthetic graphs are generated based on 8 strongly connected components, $C_0, \cdots, C_7$, each with 18, 28, 74, 120, 194, 268, 240 and 314 nodes respectively. The densities (defined as the number of edges divided by the square of the number of nodes) of these components are: 0.4722, 0.4235, 0.3629, 0.3435, 0.3280, 0.3202, 0.3218, and 0.3178, respectively. We set $\alpha = 0.9$ in our *Aug_Adj* algorithm and set τ (the number of eigen-pairs to search for) as 10.

We generate five synthetic graphs, each of which is composed of 7 components. Synth-1 contains 7 isolated components $C_0, C_2, \cdots, C_7$ with no inter-cluster edges. Synth-2 contains 7 isolated components $C_1, \cdots, C_7$ with no inter-cluster edges. The difference between Synth-1 and Synth-2 is that we purposely include a tiny component with 18 nodes (C_0) in Synth-1, which is used to demonstrate the existence of eigen-gaps would not be a reliable criteria alone to determine the eigenvectors used for clustering. Synth-3 is generated by adding inter-community edges with probabilities 0.1 between all pairs of components for both incoming and outgoing directions to Synth-2. Synth-4 is by increasing the probability between C_6 and C_7 to 0.27 whereas Synth-5 is generated by increasing the probability between C_6 and C_7 to 0.5. The clustering results are shown in Table 3, where Det indicates the number of clusters detected, M is the modularity, and Acc is the accuracy.

For Synth-1, we find that by using the naive symmetrization with AdjCl algorithm, only 6 clusters are detected. This is because the spectral radius corresponding to the perturbed Perron-Frobenius eigenvector of C_0 does not fall in the range of the largest 10 eigenvalues.

Table 3: Synthetic Data Results

	Synth-1			Synth-2			Synth-3			Synth-4			Synth-5		
Method	Det	Acc	M	Det	Acc	M	Det	Acc	M	Det	Acc	M	Det	Acc	M
Lap	7	1.000	0.761	7	1.000	0.762	6	0.801	0.362	6	0.806	**0.390**	6	**0.806**	**0.356**
Ncut	7	1.000	0.761	7	1.000	0.762	7	0.995	0.419	6	0.802	0.384	6	**0.806**	**0.356**
SpokeEn	6	0.985	0.760	6	0.997	0.761	6	0.977	0.469	4	0.724	0.380	5	0.784	0.355
AdjCl	6	0.985	0.760	7	1.000	0.762	7	**1.000**	**0.472**	4	0.724	0.380	5	0.784	0.355
Aug_Adj	7	1.000	0.761	7	1.000	0.762	7	**1.000**	**0.472**	**7**	**0.997**	0.364	6	**0.806**	**0.356**

If τ is increased to 15, the algorithm detects this eigen-pair and then can correctly identify 7 clusters. SpokeEn can only detect 6 clusters even if $\tau = 50$, which indicates that it is more susceptible to noises in the spectral space. To test our hypothesis, we increase the size of the smallest component in Synth-2. We can see AdjCl successfully identifies the correct number of components and assigns the corresponding nodes correctly to their components for Synth-2 with $\tau = 10$. However, SpokeEn fails again to detect 7 communities in this setting even if τ is increased to 50.

For Synth-3, the results are very stable for all the methods with different symmetrization techniques. In this setup, the densities of inter-community edges are smaller than those of inner-community edges, so most algorithms can find 7 components. The Lap and SpokeEn detected 6 components. It is possible that the weighted symmetrization assigned some boundary nodes to incorrect clusters. Adjacency based methods outperform the other methods due to correct selection of eigenvectors for a well conditioned adjacency matrix.

For Synth-4, components C_6 and C_7 are on the verge of merging together. Our *Aug_Adj* algorithm identifies 7 clusters and assigns most nodes correctly. We also find, when we set $\alpha = 1$, our algorithm only detects 6 clusters with modularity 0.390 and accuracy of 0.806. This result suggests that, based on an adjusted objective function, our method can detected overlapping communities. For Synth-5, components C_6 and C_7 are merged together due to

dense inter-cluster edges. Lap, Ncut and our method correctly report 6 components, while the other methods report 5.

From the above results, we have some consistent observations: methods based on matrix transformations tend to introduce both redundant information and noises that will cause graph partitions to be inaccurate; when the community sizes are not well balanced, naive symmetrization could fail to detect small communities in the adjacency eigenspace. These observations coincide with our discussions in previous sections. The results for Synth-4 lead to the speculation that the down tuned significance requirement for objective functions could lead our method to detect certain hidden structures of components. This could potentially be useful for studying micro-structures of components or overlapping problems of communities. Due to the flexibility of the objective for our algorithm, it has many potential uses for analyzing both local and global structures of a network.

3.5.2 Twitter Data

To show that the proposed algorithm is scalable and robust, we tested it on a large twitter data set. The retweet data was collected from 2013.10.23 to 2013.12.16 by using Twitter's public API. In order to have a denser graph, we removed those nodes with less than 1 incoming edge. The resulting network graph contains 5176820 nodes. The density of the reduced graph is 8.394×10^{-7}. As a result, we have a directed weighted adjacency matrix that is a good fit to test our algorithm.

We chose $\tau = 25$ and $\alpha = 1$ as the input. Ideally, we would like set τ as a large value, but it would be too time and space consuming. Therefore, we only consider the eigenpairs that correspond to the largest τ eigenvalues, since they represent those clusters with nodes

that are more influential to the network by having more incoming and out going edges. Our method then will screen out the eigenpairs that either are not corresponding to the spectral radii or do not contribute significant enough to the modularity measure.

The graph was partitioned into 16 clusters with $M = 0.347$. Figure 3 shows the nodes distribution across the clusters. Cluster 1 is the smallest with only 15 nodes. Cluster 8 is the largest and has 1855961 nodes. By checking the eigenvector corresponding to cluster 1 and the edges connecting those nodes, we confirmed that this cluster is indeed an isolated component with a nonnegative eigenvector, which coincides precisely with our theoretical assumption. Hence, the results demonstrated that the *Aug_Adj* method can detect clusters regardless of their sizes even in a very large graph by avoiding noises introduced by eigenvectors that are not Perron-Frobenius.

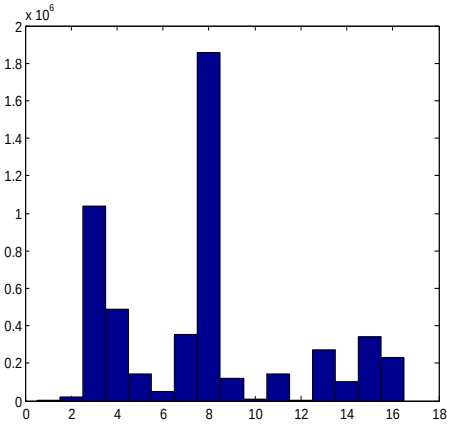


Figure 3: Histogram

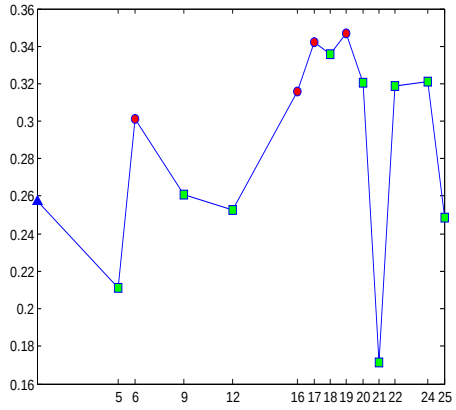


Figure 4: Modularities

Figure 4 shows the modularities of various combinations of eigenvectors. The x-axis shows the indexes of eigenvectors tested by our algorithm. The initial indexes of eigenvector candidates are: $S_0 = \{1, 2, 3, 4, 7, 8, 10, 11, 13, 14, 15, 23\}$, whose modularity is indicated by the triangle. Based on our theory, those eigenpairs corresponds to the perturbed

Perron-Frobenius simple invariant subspace. In on our algorithm, if the modularity score $M_i \geq M_{i-1}$, then the i-th eigenvector will be added to the candidate set. Therefore, further analyses included $\{6, 16, 17, 19\}$, whose modularities are indicated as circles in Figure 4. On the other hand, eigenvectors $\{5, 9, 12, 18, 20, 21, 22, 24, 25\}$ indicated by squares were removed, since they do not improve partition results.

Due to the mechanism of how the eigenvectors are selected, our algorithm can potentially reduce the time needed to optimize objective functions by only testing a subset of the real eigenvectors. As a result, computational time can be reduced, since the initial set S_0 do not need to be tested. In the worst case when S_0 is empty, our method will degrade to Modularity based method in [79] and its variants.

3.6 Summary

In this chapter, the properties of the adjacency eigenspaces of DUGs were studied. We started our work by learning from the observations in the Perron-Frobenius eigenspaces. Then we began the theoretical work from networks with disconnected communities by making the assumption that each community should be a strongly connected component. By using the matrix perturbation theory, we constructed an orthonormal basis containing the Perron-Frobenius eigenvectors corresponding to all communities to achieve the orthornormal reduction of the adjacency matrices of DUGs and described mathematically how the projections of nodes would behave in the perturbed Perron-Frobenius simple invariant subspace of an observed graph. Then, we extended our theories by replacing the original assumption of community structures with a weaker one that only requires a community to have a strongly connected core component, so that they can be used to study the

networks without clear community structures. A spectral clustering algorithm was developed and compared with other representative methods within the same domain on various synthetic data sets. The scalability and robustness of our algorithm was tested on a large Twitter data set.

For future works, the theories and algorithms can be extended to analyze various graph related problems including but not limited to: studying the microstructure of a community, analyzing the changes of the macrostructure of a network, and detecting network anomalies.

The preliminary results of this chapter were published in [65].

CHAPTER 4: SPECTRAL PROPERTIES OF DIRECTED SIGNED GRAPHS

As introduced in the previous chapter, the adjacency eigenspace of a directed unsigned network contains key information of its underlying structure. However, there has been no study on spectral analysis of the adjacency matrices of directed signed graphs (DSGs). In this chapter, we derive theoretical approximations of spectral projections from DSGs using matrix perturbation theory. We use the derived theoretical results to study the influences of negative intra cluster and inter cluster directed edges on node spectral projections. We then develop a spectral clustering based graph partition algorithm, SC-DSG, and conduct evaluations on both synthetic and real datasets. Both theoretical analysis and empirical evaluation demonstrate the effectiveness of the proposed algorithm.

4.1 Analysis of Directed Signed Graphs

In social networks, relationships between two individuals are often directed, such as Twitter following, phone calls, and voting. Directed graphs are used to capture asymmetric relationships between individuals. Relationships could have more than two status like presence or absence of a trust/friendship between two individuals. For example, they could also be negative to express distrust or dislike. Signed networks are used for this purpose. Spectral properties for Laplacian and its variants of the DUGs have been studied in the past (refer to the surveys [8,71,101]) and spectral analysis of signed undirected graphs have also been studied [138]. For example, [58] used spectral properties of signed graphs for link pre-

diction. [117] extended the modularity metric for unsigned graphs to the signed modularity for signed graphs. [60] studied the spectral properties of signed normalized Laplacian transformation from the original signed adjacency matrix and developed methods for spectral clustering, link prediction and graph structure visualization.

However, spectral analysis of the adjacency eigenspaces of DSGs has not been studied. In the ideal case of DSGs, all the intra community edges are positive and all the inter community edges are negative since the members within one community tend to hold the same opinion towards each other while members from different communities tend to dispute. However, in real world datasets such as Epinion, negative links are also present within communities and some positive links are present between communities.

It was shown in [132] and [65] that matrix perturbation theories can be used as a powerful tool for explaining the effects of inter community edges on the spectral projection behaviours of the given adjacency matrix directly. The former work provided theoretical results for undirected graphs, while the later work conducted theoretical analysis for DUGs. [131] analyzed the K-balanced undirected signed graphs by using matrix perturbation approach. [128] analyzed the effects of negative edges on the spectral properties of signed and dispute networks. However, the influences of negative edges to the spectral properties of DSGs remain unclear, so many problems in this domain are still open.

The core idea of applying the matrix perturbation theories on spectral graph analysis is to model the observed graph (with K communities) as the perturbation of intra-community edges on a K-block graph (with K disconnected communities) and study how the spectral space formed by leading eigenvectors as well as node projections in the space are changed before and after perturbation. However, when applying the matrix perturbation

theories on DSGs, one main difficulty is to deal with the complex eigenpairs associated with the asymmetric adjacency matrix. In the previous chapter and also in the work [65], we utilized the strong-connectedness property of the communities and the real Perron-Frobenius eigenvalue and eigenvector of each community, thus eliminating the need for dealing with complex eigenpairs. However, when the graphs have negative intra cluster edges, the Perron-Frobenuius eigenpairs may not exist and the eigenpair corresponding to the spectral radius may not be real any more. In this chapter, we propose to handle the inter cluster and intra cluster negative entries of DSGs.

We apply matrix perturbation theories to derive several key theoretical results for analyzing negative inter cluster edges. Our key results can answer the following important questions:

- How will the negative inter cluster edges affect the spectral projections of each node?
- Will negatively linked nodes be pushed away from each other, while positively linked nodes be pulled towards each other like those in undirected signed graphs?
- What is the role of the directionality of an edge on node spectral coordinates?
- Why can spectral projection be used for spectral clustering?

For negative intra cluster perturbation, we study how to deal with complex eigenpairs for DSGs. We explain why negative edges change real eigenpairs to complex eigenpairs. These questions are crucial in identifying the spectral properties of cluster relationships and developing spectral clustering algorithm for DSGs. Our algorithm deals with the correct selection of complex eigenpairs based on Perron-Frobenius properties, splits of complex

valued eigenvectors into real and imaginary parts, projects nodes into the spectral subspace, and applies k-means algorithm to find clusters. Our theoretical analysis based on matrix perturbation rigorously demonstrates that the perturbed Perron-Frobenius invariant subspace can indeed capture the structural properties of DSGs in the spectral domain. We emphasize that our algorithm directly identifies clusters of DSGs in the spectral space without transforming the adjacency matrices or modifying the objective functions. We conduct evaluations on several synthetic datasets and real networks and compare the accuracy results with several state-of-the-art spectral clustering methods. Results demonstrate the effectiveness of the proposed method.

4.2 Spectral Analysis of DSGs

4.2.1 Perturbation

A directed signed graph with n nodes can be represented as its adjacency matrix $\tilde{A}_{n\times n}$ with $\tilde{A}_{ij} = 1$ (-1) if there exists a positive (negative) edge pointing from node v_i to node v_j and $\tilde{A}_{ij} = 0$ otherwise. Since $\tilde{A}_{ij}$ and $\tilde{A}_{ji}$ may not have the same value, $\tilde{A}$ is asymmetric. The spectral decomposition of $\tilde{A}$ takes the form $\tilde{A} = \sum_i \lambda_i \boldsymbol{x}_i \boldsymbol{x}_i^T$. All the eigenvalues are are assumed to be in descending order in magnitude. Formally, let $\Lambda = (\lambda_1, \cdots, \lambda n)$ be the eigenvalues of matrix $\tilde{A}$, then $\rho(\tilde{A}) = max(|\Lambda|)$ is called the spectral radius of $\tilde{A}$.

We perform the spectral projections as shown in Equation (1). The basis of the spectral space are formed by eigenvectors of the given adjacency matrix. The spectral space is of full rank n, when all the eigenvectors are linearly independent. In most applications, only the first K eigenpairs contain major topological information. The row vector $\boldsymbol{\alpha}_u = (x_{1u}, x_{2u}, \cdots, x_{Ku})$ is the coordinate of node u in the spectral space. However, when

negative edges are included in DSGs, the eigenpairs could be complex.

For DSGs, negative edges could exist within communities and between communities. We treat the negative edges within each cluster as intra cluster perturbation and treat both positive and negative inter cluster edges as inter cluster perturbation. Formally, we have

$$\widetilde{A} = A + E = \begin{pmatrix} A_1 & & 0 \\ & \ddots & \\ \mathbf{0} & & A_K \end{pmatrix} + E_I + E_O \tag{14}$$

where A is a K-block matrix and each diagonal component A_i is nonnegative, E_I is a K-block matrix corresponding to intra cluster perturbation and each diagonal component E_i contains negative intra cluster edges, and E_O contains both positive and negative inter cluster edges. For each cluster C_i, its A_i is a nonnegative matrix and those entries with zero denote the absence of edges. Based on Perron-Frobenius theorem [86], for a non-negative square matrix, the largest eigenvalue (called Perron-Frobenius eigenvalue) is real and non-negative and the associated eigenvector (called Perron-Frobenius eigenvector) is unique and nonnegative. If we choose $\boldsymbol{x}_{C_i}$ to be the Perron-Frobenius eigenvectors of corresponding communities, then this spectral projection space satisfies the properties described in Lemma 6. We present our theoretical results to explain how E_O and E_I affect the associated spectral projection behaviour in Sections 4.2.2 and 4.2.3 respectively .

4.2.2 Spectral Analysis of Inter Cluster Perturbation

In this case, our model is simplified as $\widetilde{A} = A + E_O$. We studied in [65] and Chapter 3 the spectral properties of directed unsigned graphs based on the matrix perturbation theories [109] and works [107, 108]. Because the eigenvectors of asymmetric matrices do not

form an orthonormal basis naturally, we developed a method of constructing orthonormal basis and derived the approximations of the eigenvectors when treating the graph as a perturbation from a block matrix. The derived theories in [65] can be generalized to DSGs setting although although E_O contains both positive and negative inter cluster edges. This is because both A_i is nonnegative, thus having the Perron-Frobenius simple invariant subspace. We focus on how positive and negative edges in E_O affect the spectral coordinates. To be consistent, in the remaining part of Section 4.2.2, E denotes E_O.

The results of perturbed spectral space from Theorem 2 and the results of the spectral approximation from Theorem 3 from the previous chapter still hold true for DSGs, since they are derived for general graph pertrubations.

However, the work [65] associated with the previous chapter only gave the above approximation formula and did not examine how the node spectral coordinates change under perturbation of inter cluster edges. This is because the entry $\sum_{j=1}^{n} \nabla_{uj} \sum_{v \in \Gamma_u^i} \frac{e_{jv} x_{iv}}{\lambda_i}$ in the i-th column position of the spectral coordinate in Equation (11) is very complicate compared with that of undirected graphs and hence it is difficult to determine the influence of the perturbation. In this chapter, we propose a solution by decomposing the perturbation into each edge and explicitly quantifying how one single inter cluster edge $u \to v$ changes the spectral coordinates of u and v.

Without loss of generality, suppose nodes u and v are from community C_1 and C_2 respectively, there is a directed edge from u to v, $u \to v$, which could be positive or negative. Before the edge added, the spectral coordinates for nodes u and v in the two dimensional space are $\begin{pmatrix} x_{1u} & 0 \\ \mathbf{0} & x_{2v} \end{pmatrix}$. After the edge added, from Theorem 3, the spectral coordinates

are $\begin{pmatrix} x_{1u} & \nabla_{u1}\frac{e_{uv}}{\lambda_2}x_{2v} \\ \mathbf{0} & x_{2v}+\nabla_{v1}\frac{e_{uv}}{\lambda_2}x_{2v} \end{pmatrix}$.

Our next theorem shows that the change of spectral coordinates depends on both the Perron-Frobenius eigenvalue of the node's community and the edge directionality.

Theorem 5. *Denote $(\lambda_1, \boldsymbol{x}_1)$ and $(\lambda_2, \boldsymbol{x}_2)$ as the Perron-Frobenius eigenpair of C_1 and C_2 respectively. Nodes u and v are from community C_1 and C_2 respectively.*

1. *When $u \to v$ is positive,*
   (a) *If $\lambda_1 > \lambda_2$, node u has a clockwise rotation while node v stays on its original axis.*
   (b) *If $\lambda_1 < \lambda_2$, node u stays on its original axis while node v has a clockwise rotation.*
2. *When $u \to v$ is negative,*
   (a) *If $\lambda_1 > \lambda_2$, node u has an anti-clockwise rotation while node v stays on its original axis.*
   (b) *If $\lambda_1 < \lambda_2$, node u stays on its original axis while node v has an anti-clockwise rotation.*

Proof. For 1(a), node v has spectral coordinate $(0, x_{2v} + \nabla_{v1}\frac{e_{uv}}{\lambda_2}x_{2v})$. Therefore, node v will stay on its original axis. On the other hand, node u has spectral coordinate $(x_{1u}, \nabla_{u1}\frac{e_{uv}}{\lambda_2}x_{2v})$. The angle β of the spectral coordinate vector of node u with the x_1 axis will be $\arctan(\frac{\nabla_{u1}\frac{e_{uv}}{\lambda_2}x_{2v}}{x_{1u}})$. The top part $\nabla_{u1} * \frac{1}{\lambda_2}$ takes the full form as $(Y(\lambda_2 I - L_2)^{-1}Y^H)_{u1}$ as in Theorem 2. The diagonal of L_2 are the other eigenvectors of A by the

construction process. Furthermore, L_2 itself is upper triangular by Schur's Theorem. Then the diagonal entries of $(\lambda_2 I - L_2)^{-1}$ becomes $(\lambda_2 - \lambda_1, \lambda_2 - \lambda_3, \cdots, \lambda_2 - \lambda_n)^{-1}$. If we divide ∇ by the term $(\lambda_2 - \lambda_1)^{-1}$ and relabel it as ∇^*, then the spectral coordinates of u becomes $(x_{iu}, (\lambda_2 - \lambda_1)^{-1} \nabla^*_{u1} e_{uv} x_{2v})$. Then the angle β becomes $\arctan(\frac{(\lambda_2 - \lambda_1)^{-1} \nabla^*_{u1} e_{uv} x_{2v}}{x_{1u}})$. Since $\lambda_2 - \lambda_1 < 0$, β will be a negative angle, which indicates that node u will rotate clockwise to the fourth quadrant.

For 1(b), if $\lambda_1 < \lambda_2$, by relabeling ∇^* if necessary, the angle β takes the same equation as $\arctan(\frac{(\lambda_2 - \lambda_1)^{-1} \nabla^*_{u1} e_{uv} x_{2v}}{x_{1u}})$. Since $\lambda_1 < \lambda_2$, β will be a positive angle, which indicates that node u will rotate counter-clockwise and the vector will remain in the first quadrant.

For 2(a), $e_{uv} < 0$ and $\lambda_1 > \lambda_2$. By relabeling ∇^* if necessary, the angle β takes the same equation as $\arctan(\frac{(\lambda_2 - \lambda_1)^{-1} \nabla^*_{u1} e_{uv} x_{2v}}{x_{1u}})$. Since $\lambda_1 > \lambda_2$, then $(\lambda_2 - \lambda_1)^{-1} e_{uv}$ will be positive, which indicates that node u will rotate counter-clockwise and the vector will remain in the first quadrant.

For 2(b), $e_{uv} < 0$ and $\lambda_1 < \lambda_2$. $(\lambda_2 - \lambda_1)^{-1} e_{uv}$ will be negative, which indicates that node u will rotate clockwise and the vector will be in the fourth quadrant.

□

Illustrative Example. Figure 5 shows a toy graph where different edge types will be added between nodes 8 and 25. Figure 6 illustrates the rotations with respect of perturbation edge directions and signs in the spectral space. The triangles represent nodes from cluster C1, labeled with 1-8 and 15 in Figure 5, while the crosses represent nodes from cluster C2, labeled with 16-25. Node 8 is marked with green and node 25 is marked with magenta in order to separate from other nodes. The Perron Frobenius eigenvalue is 1.8839 for cluster

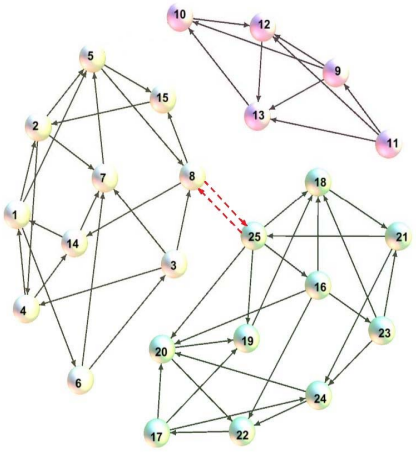
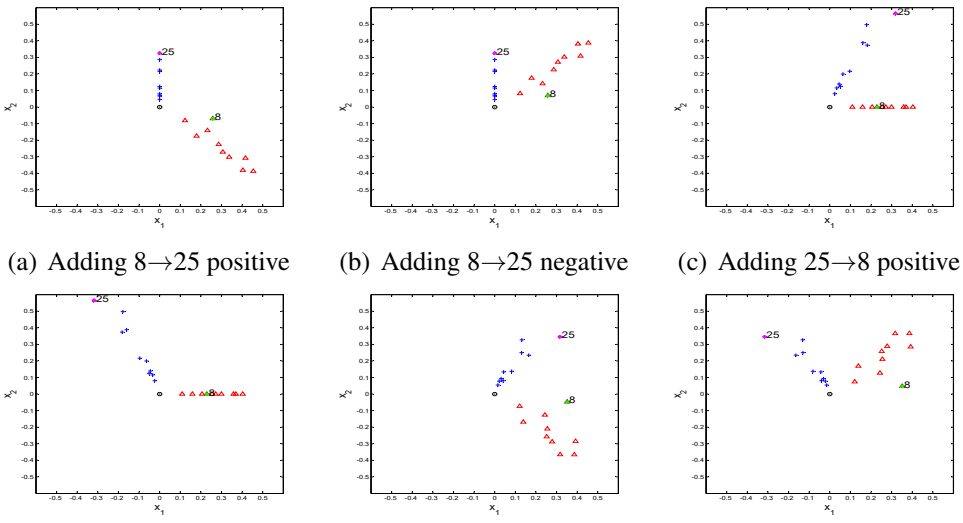


Figure 5: Example graph with 3 communities, where node 8 and 25 are connected by negative or positive edges.

C1 and 1.7284 for cluster C2, so that $\lambda_1 > \lambda_2$. The sub-figures on the left hand side correspond to positive perturbation, with edge 8→25, 8←25 , and 8↔25, and those on the right side correspond to the negative perturbation respectively. All the observations match our results in Theorem 5. For example, Figure 6(a) shows node 8 and other nodes in C1 rotate clockwisely while node 25 and other nodes in C2 stay on the original line with a positive edge 8→25, matching our result 1(a) in Theorem 5. Similarly, Figure 6(b) shows node 8 and other C1 nodes rotate anti-clockwisely with a negative edge 8→25, matching our result 2(a) in Theorem 5. Figures 6(c) and 6(d) show the effect due to edge directionality. Figures 6(e) and 6(f) show the combined effects of both directions.

4.2.3 Spectral Analysis of Intra Cluster Perturbation

In general, the subgraph for each cluster is treated as an intra cluster perturbation from a nonnegative subgraph A_i such that $\widetilde{A_i} = A_i + E_i$, with E_i containing all negative intra cluster edges. The intra cluster perturbation $\widetilde{A_i} = A_i + E_i$ where E_i containing all negative intra cluster edges can be treated as a transition from a nonnegative graph A_i with the



(a) Adding 8→25 positive (b) Adding 8→25 negative (c) Adding 25→8 positive

(d) Adding 25→8 negative (e) Adding 8↔25 positive (f) Adding 8↔25 negative

Figure 6: Spectral Coordinates of Nodes under Perturbation with Positive or Negative Edges

Perron Frobenius property into a signed graph with uncertain properties. Depending on the amount and locations of negative edges added, the Perron Frobenius property may not hold.

Definition 2. *The characteristic polynomial of a* n *by* n *matrix* A *takes the general form:*

$$F(\lambda) = a_1 * \lambda^n + a_2 * \lambda^{n-1} + \cdots + a_n * \lambda + a_{n+1}, \tag{15}$$

where the roots for $F(\lambda) = 0$ *will be the eigenvalues of* A*.*

Those negative entries within clusters may cause those Perron roots for their characteristic polynomials to change drastically and the corresponding eigenvectors and spectral projections will also change accordingly. Since the coefficients of the polynomial in Equation (15) are determined by the determinant $|A - \lambda * I|_{det}$, which is calculated iteratively with the entries of A, the polynomial itself could be either increasing, decreasing, concave or convex. Therefore, the resulting eigenvalues and eigenvectors could be complex. How-

ever, we do not have any explicit results to show how small the negative entries should be and/or where those negative entries should locate in order for a graph to retain the Perron Frobenius property.

The theoretical results of graph theories pertaining the spectral properties of general signed graphs are relatively scarce. From the first proposition of the Perron Frobenius theorem for nonnegative irreducible square matrices in 1912 [35] to the recent works [34,86,137] that extended the result into eventually irreducible nonnegative directed graphs a few decades ago, many problems related to signed graphs remain open. There are some works on studying the relationship between graph topology and complex eigenpairs in applied mathematics and linear algebra. The authors in the work [57] pointed out that three properties can be read off the complex eigenvalues: whether a graph is nearly acyclic, whether a graph is nearly symmetric, and whether a graph is nearly bipartite. If a directed graph is acyclic, its adjacency matrix is nilpotent and therefore all its eigenvalues are zero [26]. The complex eigenvalue plot can therefore serve as a test for networks that are nearly acyclic. When a directed network is symmetric, the adjacency matrix A is symmetric and all its eigenvalues are real. As a result, a directed network close to symmetric has complex eigenvalues near the real line. Additionally, the eigenvalues of an undirected bipartite signed graph are symmetric around the imaginary axis, so the amount of symmetric along the imaginary axis can serve as an indicator for bipartivity.

4.2.4 Spectral Clustering Algorithm for DSGs

The results from Section 4.2.2 described how node spectral coordinates are changed due to inter cluster perturbation, while the results from Section 4.2.3 described the potential

complex eigenpairs due to negative intra cluster perturbation. With the two results combined, we present our spectral clustering based graph partition algorithm, SC-DSG, for DSGs.

Algorithm 2 *SC-DSG*: Spectral Clustering for DSGs

Input: A, τ
Output: cluster result K, CL

1: Compute eigenvectors of A corresponding to the τ largest eigenvalues in modulii and choose only eigenvectors with positive real part of eigenvalues, denote the set D;
2: Normalize the eigenvectors $\bar{\boldsymbol{\alpha}}_u = \frac{\boldsymbol{\alpha}_u}{\|\boldsymbol{\alpha}_u\|}$;
3: $C \leftarrow$ real eigenvectors from the set D with same signed components;
4: $K \leftarrow Cardinality(C)$, $M \leftarrow -\inf$;
5: **for** each $c \in D \setminus C$ **do**
6: **if** c is complex **then**
7: $c \leftarrow$ split into [Re(c), Im(c)]
8: **end if**
9: Apply k-means algorithm on $C \cup c$ to get clustering result CL_{temp} of K clusters;
10: Compute the signed modularity score M_{temp};
11: **if** $M_{temp} \geq M$ **then**
12: $C \leftarrow C \cup c$, $M \leftarrow M_{temp}$;
13: $K \leftarrow K + 1$, $CL \leftarrow CL_{temp}$;
14: **end if**
15: **end for**
16: Return number of clusters K and clustering result CL;

Algorithm 2 includes the following major steps: computing eigenvectors of the adjacency matrix; normalization of the eigenvectors; selecting the initial set of eigenvectors with same signed components; splitting complex eigenvectors into real and imaginary parts; projection of the nodes onto a unit sphere; clustering the nodes according to their location on the unit sphere using the classic k-means clustering algorithm; screen all the potential eigenpairs based on the signed modularity to find meaningful partitions.

Spectral clustering based partition algorithms require one to find a correct set of eigenvectors for spectral projection. This leads to the search for a set of eigenvectors that can

capture the structural information in the spectral domain. Our algorithm uses the stepwise forward strategy to find a set of eigenvectors. Eigenvectors are ordered according to the modulii of eigenvalues and only eigenpairs with positive real part are chosen for spectral clustering. We exploit Perro-Frobenius properties and include the candidate eigen-pairs based on whether they can help increase the signed modularity [7],

$$Q_s = \frac{1}{2m} \sum_{i,j \in V} (A_{ij} - \frac{d_i^+ d_j^+}{2m} + \frac{d_i^- d_j^-}{2m}) \delta(C(i), C(j)), \tag{16}$$

where d_i^+ (d_i^-) denotes the node i's positive (negative) degree and $C(i)$ denotes the node i's community.

A key step is to deal with potentially complex eigenvectors. Our algorithm uses eigenvectors corresponding to the spectral radii of each component. However, due to negative intra cluster perturbations, the eigenpairs may not be real anymore. The k-means clustering used in most of the spectral based clustering methods could not produce meaningful results in the coordinate space of $\mathbb{C}^n$, since the Euclidean distance of two complex coordinates with only imaginary part will be negative. In our algorithm, we split each complex eigenvector into the real and imaginary parts. As a result, the complex spectral coordinate space is transformed to the real space that combines both real and imaginary parts. We emphasize that both real and imaginary parts contain information for clustering, as shown in our theoretical analysis and experimental results. It is worth pointing out that we may use some distance functions defined over complex-valued vectors rather than split. After determining the eigenvectors, we project nodes into the spectral space and then apply the k-means method.

Our algorithm is developed rigorously based on our theoretical results. Our Theorem 5

shows the change of spectral coordinates depends on the Perron-Frobenius eigenvalues of the communities and the sign and directionality of the inter-cluster edge, which lays out the theoretical foundation of spectral clustering algorithm. The spectral analysis of intra-cluster perturbation shows why the Perron-Frobenius eigenpair may have complex values, which provides theoretical justification of its split into real and imagery parts in algorithm.

The calculation of the eigenvectors of an $n \times n$ matrix takes in general a number of operations $O(n^3)$, which is almost inapplicable for large networks. However, in our framework, we only need to calculate the first K eigen-pairs, which can be determined by examining the eigen-gaps [109]. Furthermore, adjacency matrices in our context are usually sparse. Therefore, the Arnoldi/Lanczos algorithm [39], which generally needs $O(n)$ rather than $O(n^2)$ floating point operations at each iteration, can be applied to calculate the most significant eigenpairs. In our implementation, we conduct eigen decomposition using Matlab's eigs() function where the Arnoldi/Lanczos algorithm is realized through the APPACK package.

4.3 Empirical Evaluation

4.3.1 Baseline Algorithms

We compare our SC-DSG with the following state-of-the-art baseline algorithms: 1) The Augmented_ADJ (AugAdj) [65] is an adjacency based spectral clustering method for directed unsigned graph; 2) UniAdj [128] is an adjacency based spectral clustering method for signed undirected graphs; 3) The signed normalized cut (SNcut) [60] is an improved version of the signed Laplacian method where weighting schemes are adjusted to form better partitions; 4) SC-DSG-M is a variant of SC-DSG and only uses the modulii of the

eigenvector entries as spectral coordinates; and 5) SC-DSG-Re is another variant of SC-DSG and only uses the real part of the eigenvector entries as spectral coordinates. The two variants of SC-DSG are used to demonstrate the usefulness of incorporating the whole complex eigenvectors in the clustering. AugAdj can be used to deal with the DSGs as it ignores the use of any complex eigenpairs. Both UniAdj and SNcut require symmetric adjacency matrices as input. In our experiment, we build the symmetrized versions of the original directed graphs by the following process: $A_{ij} = A_{ji} = -1$ if either$A_{ij} = -1$ or $A_{ji} = -1$, $A_{ij} = A_{ji} = 1$ if either $A_{ij} = 1$ or $A_{ji} = 1$, and $A_{ij} = A_{ji} = 0$ otherwise. We limit the search for each method to 50 eigenpairs. Signed modularity, DBI and average angle between clusters in the spectral projection space are reported in addition to accuracy.

Table 4: Statistics of synthetic data and partition quality

Dataset	Edge/+ratio/−ratio		k	DBI	Q	Angle	ACCURACY(%)					
	Intra	Inter					SC-SDG	SC-DSG-M	SC-DSG-Re	AugAdj	UniAdj	SNCut
Syn-1	67653/0.4/0	144283/0.2/0	5	0.1745	0.2770	89.3°	100	100	100	100	100	100
Syn-2	67588/0.4/0	144335/0.1/0.1	5	0.4711	1.9141	92.2°	100	100	100	100	100	100
Syn-3	67545/0.4/0	144362/0/0.2	5	0.0926	-1.0798	90.1°	100	100	100	100	100	100
Syn-4	67618/0.4/0	400420/0.7/0	4	1.9774	0.0321	76.5°	**72.9**	68.3	68.8	70.3	71.9	62
Syn-5	80749/0.4/0.08	144294/0.2/0	5	0.4290	0.2221	87.9°	100	100	100	100	100	100
Syn-6	81019/0.4/0.08	144372/0/0.2	5	0.3827	0.4442	89.1°	100	100	100	100	100	100
Syn-7	80789/0.4/0.08	438193/0.4/0	4	1.4309	0.0776	82.0°	**92**	**92**	**92**	**92**	**92**	89.9
Syn-8	101002/0.4/0.16	144220/0.2/0	5	1.9958	0.0749	76.8°	**67.5**	**62.9**	**62.9**	**62.9**	**62.9**	65.1
Syn-9	127448/0.4/0.36	144283/0.1/0.1	5	2.9701	-0.2260	82.4°	**59.3**	57.1	58.9	n/a	55.1	56

4.3.2 Synthetic Data

We generate 9 synthetic graphs. Each graph has 5 clusters with 240,220,200,180 and 160 nodes. The edges for Syn-1 to Syn-9 are generated using uniform random distribution. The column "Intra" of table 4 shows the number of intra cluster edges as well as the positive and negative densities. In particular, the $x/y/z$ means that there are x intra cluster edges and the density of positive (negative) edges is $y \times 100$ ($z \times 100$) percent. Similarly, the column "Inter" shows the corresponding statistics for inter cluster edges. The negative edges for both intra and inter clusters are injected into the 5-block graph so that the perturbed graphs

possess desired structural properties.

For Syn-1 to Syn-4, there are 40% positive edges but no negative edges within clusters. The inter cluster positive edge density of Syn-1 is 0.2 and there is no negative inter cluster edge. All the methods achieved 100% accuracy. The average cluster angle is close to 90 degrees. In Syn-2, half of the inter cluster edges from Syn-1 are converted into negative edges. The perturbed Perron Frobenius invariant subspace contains the real eigenvectors corresponding to the spectral radii of the clusters. The average angle between clusters is 92.2 degrees. In Syn-3, all the inter cluster edges are negative. All methods achieve 100 percent accuracy and the average angle is 90.1 degrees. In Syn-4, the inter cluster positive edge density is increased to 0.7 without the inter cluster negative edge. In this setting, all methods report 4 clusters, where the accuracies drop to around 60 to 70 percent. Since we have dense inter cluster connections, the results are expected.

For Syn-5, Syn-6 and Syn-7, the positive (negative) intra cluster edge density is 0.4 (0.08). For Syn-5, the positive inter cluster density is 0.2 with no negative inter cluster edge. All methods achieve 100 percent accuracy. For Syn-6, the negative inter cluster edge density is 0.2. Since the inter cluster contains only negative edges, all methods still achieve 100 percent accuracy. For Syn-7, more positive inter cluster edges are added. The partition accuracy drops. Same as Syn-4, only 4 clusters are detected. Our SC-DSG achieves the best accuracy.

For Syn-8, the negative intra cluster perturbation is doubled to 0.16. The positive intra cluster edge density remains as 0.4. The positive inter cluster edge density is set to be 0.2. The accuracy values drop by over 20 percent for all methods. There exist some complex eigenvalues whose modulii equal the spectral radius. For Syn-9, the negative intra cluster

edge density is further increased to 0.36. Both the positive and negative inter cluster density is 0.1. This is the most complex case for DSGs. With no surprise, all clustering methods perform poorly, with SC-DSG achieving the best accuracy (59.3%).

To summarize, when under small inter cluster perturbations (as the cases for Syn-1 to Syn-3, Syn-5 and Syn-6), as long as clusters satisfy the Perron Frobenius property, all methods perform the same, since the correct perturbed Perron Frobenius simple invariant subspace is captured by all methods. As demonstrated in Syn-4 and Syn-7, dense inter cluster edges cause clusters to merge, so the clustering accuracies decrease. As demonstrated in Syn-8 and Syn-9, when the Perron Frobenius property begins to disappear, the clustering accuracy will decrease more. In all cases, SC-DSG achieves the best accuracy.

4.3.3 Real Data

Table 5: Real Data Statistics and Results

Algorithm	NETWORK STATISTICS			SIGNED MODULARITY/CLUSTERS			
	Nodes	+Edges	-Edges	SC-DSG	AugAdj	UniAdj	SNCut
Sampson's	18	97	87	**2.52**/3	1.4200/2	-0.8757/15	-1.0503/11
Slashdot	79120	370234	117517	**0.16**/6	0.1512/4	0.155/5	0.1072/22
Wikisigned	138592	650653	89744	**0.1734**/5	0.0785/3	0.0848/5	0.0789/37
Epinion	131828	717667	123705	**0.3416**/5	0.337/6	-0.174/5	0.2595/13

In this section we conduct our empirical experiments on four real DSGs, Sampson's, Wikisigned, Slashdot Zoo and Epinion. Sampson's work [94] contains the opinions of 18 trainee monks about their relationships towards each other during the period of time when the clique fell apart. Each monk was asked to rate others from 1 to 3 based on like or dislike. Later on, the responses were converted into an binary signed adjacency matrix. Slashdot [59] is a technology news site where users can mark others as "friend" or "foe" and influence scores seen by them. Therefore, the entire network could be seen as a trust

network. Wikisigned [1] contains interactions between the users of the English Wikipedia that have edited pages about politics. Each interaction, such as text editing, reverts, restores and votes are given a positive or negative value. Epinion [63] is an online product rating website. The users can choose to trust or distrust others and self vote is allowed. As a result, the network could be viewed as a trustworthy relationship network.

Table 5 shows the graph statistics, the signed modularity and number of clusters reported by each method. We see that our method achieves the best signed modularity value for all four datasets. We observe that the eigenvector associated with cluster 6 is complex for Slashdot, the eigenvector associated with cluster 4 is complex, all the others are real. Note that we cannot report accuracy because of no ground truth about these four datasets.

4.4 Related Work

There is a large literature on spectral analysis of the graph Laplacian or normal matrix for unsigned networks with various applications such as spectral clustering and graph visualization. Refer to the survey [125]. These spectral clustering methods exploit a basic fact in spectral graph theory that the number of connected components in an undirected graph is equal to the multiplicity of the eigenvalue zero in the Laplacian matrix of the graph. In spectral analysis of the Laplacian matrix or the normal matrix, the coordinates are arranged to make the sum of all the distance between two nodes smallest. In their projection spaces, closely related nodes are pulled together to form clusters. Several works (e.g., [10, 61, 89]) have applied matrix perturbation theory to analyze spectral techniques and gave theoretical justification. In [61], the authors provided a theoretical explanation why the bottom k eigenvectors of the Laplacian matrix can be used for graph partition. Different from the

Laplacian matrix or normal matrix, the properties of the adjacency eigenspace have only received attentions in some recent work including the EigenSpoke pattern [91] of sparse graphs and the orthogonal line pattern [132–134] for k-block networks.

Mining signed network attracts increasing attention [115]. Research works [21, 51, 60] are based on balance theory [44] which can be viewed as a model of likes and dislikes. For example, in [51], the authors showed that the stability of sentiments is equivalent to k-balanced graphs. The authors in [131] conducted the spectral analysis of approximate k-balanced signed graphs by applying matrix perturbation. However, their results are only applicable for a special type of signed networks, i.e., k-balanced networks where negative connections only exist across communities and positive connections only exist inside communities. In [63], the authors studied signed networks based on status theory where a positive directed link indicates that the creator of the link views the recipient as having higher status and a negative directed link indicates that the recipient is viewed as having lower status. Researchers also extended some of those existing measures and clustering algorithms for unsigned graphs to signed graphs. Several notable works include the extension of modularity on signed graphs [117], the spectral clustering based on the signed graph Laplacian [60]. However, they failed to clearly relate the structures in signed graphs with patterns in the spectral space.

Researchers have developed approaches and algorithms to deal with the clustering in directed graphs because relationships in many networks are asymmetric. Refer to [71] for a recent survey. Roughly speaking, they can be classified into two categories. In the first category, the directed graph is converted into an undirected one, either unipartite or bipartite, where edge direction is preserved, e.g., via edge weights of the produced unipar-

tite graph [95] or edges in the produced bipartite graph [143]. Clustering algorithms for undirected weighted graphs are then applied. Methods in the second category are mainly based on the idea of extending clustering objective functions and methodologies to directed networks. In those approaches, the graph clustering is expressed as an optimization problem and the desired clustering properties are captured in the modified objective criterion. For example, researchers developed the directed versions of modularity [83], the objective function of weighed cuts in directed graphs [72], and the spectral graph clustering based on the Laplacian matrix of the directed graphs [25, 142]. However, it is unclear to what extent the information about the directionality of the edges is retained by these approaches. In [65], we studied to directly analyze the spectral properties of the adjacency matrix of the underlying directed network. When the concern is with directed graphs, one main difficulty for spectral clustering is to deal with the complex values for eigenpairs associated with the asymmetric adjacency matrix. The approach utilizes the connectedness property of the components of a network to screen out irrelevant eigenpairs and the Perron-Frobenius eigenpairs are all real, thus eliminating the need for dealing with complex eigenpairs. However, that approach cannot be used for DSGs because the perturbed Perron-Frobenius eigenpairs are complex valued.

4.5 Summary

In this chapter, we conducted spectral analysis of DSGs. Spectral methods have been successfully adopted in solving graph or network structure related problems. However, most work focus on spectral analysis of undirected unsigned graphs or transform underlying directed graphs into symmetric matrices like Laplacian. To our best knowledge, our

work is the first to study the complex-valued eigenvalues/eigenvectors for clustering DSGs. By using matrix perturbation theory, we derived the approximations of the spectral coordinates of nodes in the spectral projection space formed by perturbed Perron Frobenius invariant subspace and explained the effects of added intra and inter cluster edges to the spectral coordinates. A spectral clustering algorithm for DSGs, SC-DSG, was proposed according to the theoretical results and was tested on both synthetic and real datasets. The results demonstrated the effectiveness of the algorithm.

Many future research topics could be built upon our current theoretical framework, such as fraud detection, dynamic network analysis, and signed network embedding. We will also study the scalability of our algorithm. We would emphasize that our algorithm has the same bottleneck, the eigen decomposition, as all other spectral clustering methods.

Repeatability. Our software together with the datasets used in this chapter are available at `https://github.com/gnemeuyil/DSG`.

The preliminary results of this chapter is published in [66].

CHAPTER 5: SOCIAL NETWORK DOMINANCE BASED ON ANALYSIS OF ASYMMETRY

We focus on analysis of dominance, power, influence—that by definition asymmetric—between pairs of individuals in social networks. We conduct dominance analysis based on the canonical analysis of asymmetry that decomposes a square asymmetric matrix into two parts, a symmetric one and a skew-symmetric one, and then applies the SVD on the skew-symmetric part. Each individual node can be projected as one 2-dimensional point based on its row values at each pair of successive singular vectors. The asymmetric relationship between two individuals can then be captured by areas of triangles formed from the two points and the origin in each 2-dimensional space. We quantify node dominance (submissive) score based on the relative position of the node's coordinate from coordinates of all other nodes it dominates (subdues) in the projected singular vector spaces. We conduct dominance/submissiveness analysis for several representative networks including perfect linear orderings, networks with tree structure, and networks with random graphs and examine the departures of a real social network from those representative graphs. Empirical evaluations demonstrate the effectiveness of the proposed approach. The theoretical results in this chapter works for both DUGs and DSGs, so when directed graphs are mentioned, they represent general directed graphs.

5.1 Analyzing the Dominance Structure of Network

For undirected networks, researches have developed various measures to indicate the structure and characteristics of the network from different perspectives. Various properties including its size, density, power-law degree distributions, average distance, small-world phenomenon, clustering coefficient, randomness, community structures etc. have been discovered [27, 77, 110, 134]. For directed networks, researchers have developed methods to discover underlying community structure, authority ranks of individual nodes, and directions of information flow among clusters [9, 25, 53, 62, 72, 76, 79, 122, 141, 142] because relationships in many networks are asymmetric.

In directed social networks, each individual node tends to contain some amount of dominance and some amount of submissiveness. Consider an organizational network where each node denotes an individual and an edge between two nodes denotes a reporting-to relation between two individuals, the hierarchical level of an individual is determined by two scores: dominance score in terms of both how many others he dominates and who those others are, and submissiveness score in terms of both how may others dominate him and who those others are. The amount of dominance versus submissiveness at the node level can clearly affect various properties of a social network. Although dominance/submissiveness relationships play an important role in understanding the geometry and topology of social networks, very few studies have formally investigated this issue.

In this chapter, we first develop measures of dominance and submissiveness for each node. Our measures are derived from the *canonical analysis of asymmetry* originally developed in [40]. The analysis of asymmetry approach decomposes a square asymmetric matrix

X into two parts, a symmetric one, $\frac{1}{2}(X + X^T)$, and a skew-symmetric one, $\frac{1}{2}(X - X^T)$, and then applies the SVD on the skew-symmetric part. After the SVD, each individual node can be projected as one 2-dimensional point based on its row values at each pair of successive singular vectors. The asymmetric relationship between two individuals can then be captured by areas of triangles formed from the two points and the origin in each 2-dimensional space. In our work, we quantify node dominance (submissiveness) score based on the relative position of the node's coordinate from coordinates of all other nodes it dominates (subdues) in the projected singular vector spaces. The position of each individual is determined in terms of both of how many others it dominates and how important those others are. Each individual's position is determined by taking the total pattern into account. We define the authority score of a node by subtracting its submissiveness score from its dominance score. We compare our proposed authority score measure with some traditional measures such as PageRank.

We examine the use of the canonical analysis of asymmetry for several representative types of networks including perfect linear orderings, networks with tree structure, and networks with random graphs. It was well known that the projected points form a perfect arc around the origin for a network with a linear order among all individual nodes [40]. However, there are no study of the distribution of dominance/submissiveness scores of other representative graphs. For a given social network, it may lie between the network with perfect linear orderings, network with tree structure, and random graph. To determine whether a general social network is similar to one representative graph, we propose the use of Kolmogorov-Smirnov test. We also develop relative dominance (submissiveness) measures to quantify its departure from the representative graph. The empirical evaluation

over a global products and goods trade data show the effectiveness of our approach.

5.2 Dominance Framework

In this chapter, we assume the edges in the observed graph X are homogenous and represent the same type of relationship across the entire network. Depending on the problems being studied, each entry $X_{i,j}$ could be binary, denoting the presence/absence of a directed relationship like phone call from individuals i to j, or weighted, capturing the quantity of the directed relationship like the number of phone calls from individuals i to j.

5.2.1 Node Dominance/Submissiveness Measures

For any pair of individuals p and q, we propose the use of asymmetry to quantify the dominance/submissiveness between nodes p and q. Based on the canonical analysis of asymmetry [40], the spectral coordinate of each node can be viewed as projections in a set of bimensions. The number of bimensions K can be justified by the adequacy of the retained singular values expressed as a proportion of the sum-of-squares of all the singular values.

In the k-th bimension, the spectral coordinate of p is $(U_{p,2k-1}\sigma_{2k-1}^{1/2}, U_{p,2k}\sigma_{2k}^{1/2})$. Note that $\sigma_{2k-1} = \sigma_{2k}$ for each k. We use $\overrightarrow{p_k}$ ($\overrightarrow{q_k}$) to denote the vector from the origin to node p's (q's) spectral coordinate in the k-th bimension. The asymmetry between nodes p and q is defined as:

$$\begin{aligned} d(p,q) &= \frac{1}{2}\sum_{k=1}^{K} \|\overrightarrow{p_k} \times \overrightarrow{q_k}\| \\ &= \frac{1}{2}\sum_{k=1}^{K} |U_{p,2k-1} \times U_{q,2k} - U_{p,2k} \times U_{q,2k-1}| \times \sigma_{2k-1}. \end{aligned} \tag{17}$$

Note that $\frac{1}{2}\|\overrightarrow{p_k} \times \overrightarrow{q_k}\|$ is the area of the triangle formed by the spectral coordinates of p, q

and the origin in the k-th bimension. The value of $d(p, q)$ is simply the sum of the triangle areas in K bimensions.

When studying issues such as power, influence, dominance in social networks, we need to distinguish between senders and receivers. In some situations, such as being nominated as officer by students, or being the targets of advice-seeking among colleagues, receivers often occupy superior positions in a network, as opposed to the senders who often take lower positions. However, in some other situations, the reverse is true that the senders occupy superior positions, and receivers less lofty situations. For example, in international trade, the nations that send large quantity of goods to other countries are in economically more dominant position than the receiving nations. Recall in the skew-symmetric matrix, $z_{p,q} = \frac{1}{2}(x_{p,q} - x_{q,p})$. Hence, $z_{p,q} < 0$ means the dominance of p over q in the former scenario whereas $z_{p,q} > 0$ means the dominance of p over q in the latter scenario.

In a given network, any node can be viewed as an individual that resides in some tier of the hierarchy system. Therefore, by calculating the asymmetry between p and all its dominated neighbours, we define the **dominance score** of node p as:

$$DS_p = \sum_{q \in D_p} d(p, q), \tag{18}$$

where D_p contains all nodes q dominated by node p. Similarly, we define the **submissiveness score** of node p as:

$$SS_p = \sum_{q' \in S_p} d(p, q'), \tag{19}$$

where S_p contains all nodes q' dominating node p.

Dominance/submissiveness measures can be used to describe the position of the orga-

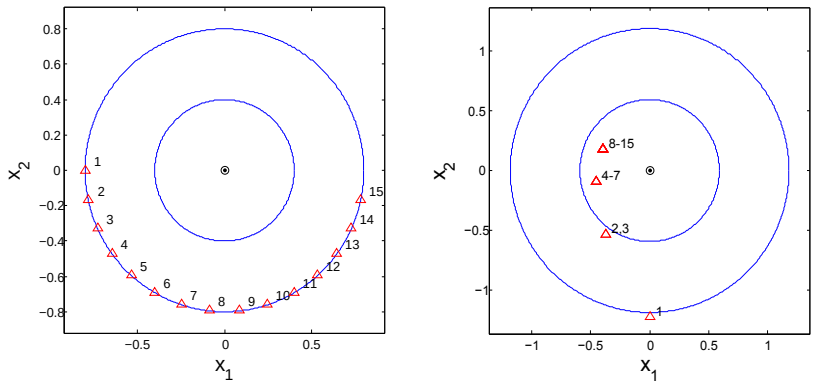
nizational hierarchy or the authority among individuals in a given network. For example, if an individual has high D_p and low S_p, he would reside on a relatively high tier of the organizational hierarchy or have high authority score. We hence define the **authority score** of node p as:

$$\Gamma_p = DS_p - SS_p = \sum_{q \in D_p} d(p,q) - \sum_{q' \in S_p} d(p,q'). \tag{20}$$

5.2.2 Graph Dominance Analysis

We start our analysis from some representative networks including graphs with perfect linear orderings, graphs with tree structure, and random graphs. For a given social network, it may lie between the graph with perfect linear orderings, that with tree structure, and random graph. We develop relative measures to quantify the departures of a general social network from those special graphs in terms of dominance/submissiveness relationships.

Graph with Perfect Linear Ordering. A total linear ordering is a binary relation that is reflexive, antisymmetric, transitive and total. For such an ordering, as a simplest example, one individual p in some sense dominates everyone, then the next q dominates everyone but p, and so on, until the last individual is dominated by all the other individuals. Formula (21) shows a linear order matrix according to the hierarchy of a network organization. In this example, $X_{i,j} = 1$ if node i dominates node j, otherwise $X_{i,j} = 0$. No two individuals



(a) Perfect linear order (b) Graph with hidden tree structure

(c) Tree graph (d) Random graph

Figure 7: Spectral Projection of the First Bimension for 15-node Graphs

have the same rank in the organizational hierarchy.

$$X = \begin{pmatrix} 0 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 1 \\ 0 & 0 & 0 & \ddots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{21}$$

Figure 7(a) shows the spectral projection for an illustrative example with 15 nodes following Formula (21). According to the work [40, 41], when the skew-symmetric matrix derived form a linear dominance matrix is subjected to SVD, the points are equidistant from one another and they are all arranged on an arc of a circle that is centered at the

origin. That circle has a radius of $2\sqrt{(\sigma_1/n)}$ where n is the number of points in the linear order. In particular, the first pair of singular values are $\sigma_1 = \sigma_2 = 1/2tan\{\frac{(n-1)\pi}{2n}\}$. When the perfect linear ordering network structure exists, the projections lie on an arc and equally divide the half circle. Therefore, each node dominates the immediate next one by $\delta = 0.5*R*|sin(180/n)|$, where R is the radius of the circle that equals to the length of the vector. We then have $DS_i = \sum_{j=i}^{n-1} \delta * j$. Similarly, we have $SS_i = \sum_{j=1}^{i-1} \delta * (i-j)$. Hence we have $\Gamma_i = \sum_{j=i}^{n-1} \delta * j - \sum_{j=1}^{i-1} \delta * (i-j)$. However, for other types of reprensentative graphs, we do not have explicit formula for the dominance score.

Graph with a Hidden Complete Binary Tree Structure. A tiered organizational hierarchy can be represented by a tree with the root node representing the highest authority and the leaf nodes representing the lowest authority. The dominance network is constructed as: $X_{i,j} = 1$ for all nodes j in the subtree with node i as its root. Figure 7(b) shows the spectral projection of the skew symmetric part of a 15-node graph with a complete binary tree [2] hierarchical structure. This example simulates an organization with 1 CEO, 2 department managers, 4 group leaders, and 8 employees. The CEO, department managers, group leaders, and employees form four tiers of the hierarchy. We can see all the projections fall on one side of the half circle and orient sequentially along a curve. As the tier gets lower, the projections get closer to the origin. The plot accurately indicates that the network contains four tiers. Node 1 has the highest degree, thus is the most distant from the origin. It belongs to the first tier. Nodes 2 and 3 reside on the second tier. Nodes 4 to 7 are on the third tier. Nodes 8 to 15 form the fourth tier, thus are the closest to the origin.

[2] A binary tree is complete if all levels except possibly the last are completely full, and the last level has all its nodes to the left side.

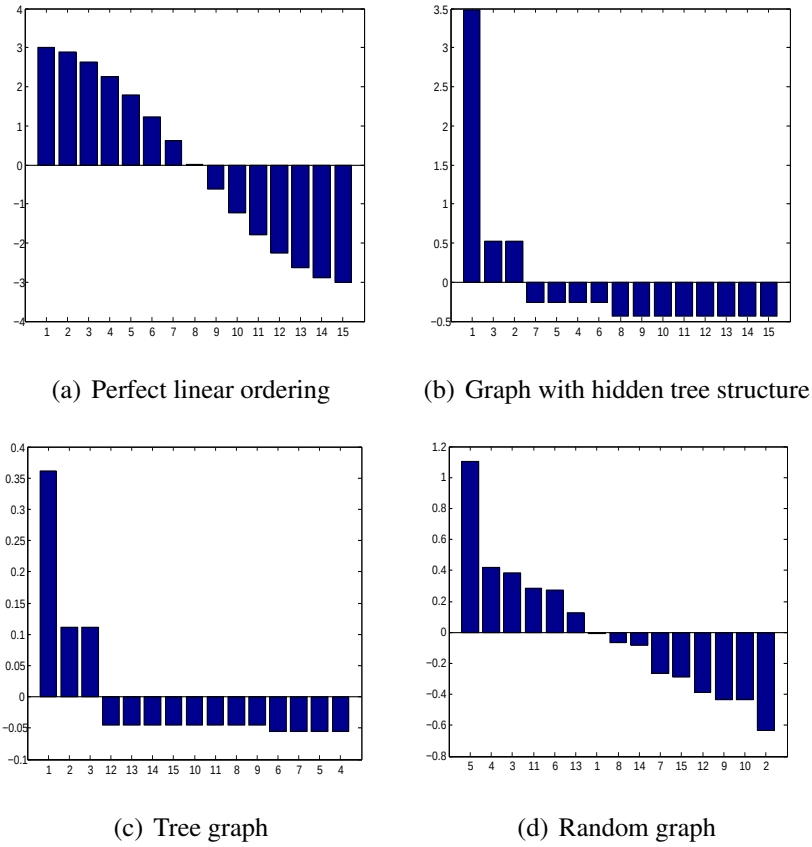
Tree Graph. When the observed graph itself has the tree structure, Figure 7(c) shows its projection. The dominance direction is counter clockwise in this plot. Node 1 dominates 2 and 3. Since the angle of vectors formed by nodes 2 and 3 is 0, there is not any dominance relationship between them. The next tier of the hierarchy contains nodes 4 to 7 that overlap on the same spot in the plot. This indicates that those nodes have the same authority status. Nodes 8 to 15 locate on the bottom of the hierarchy and have the same authority status, so their projections overlap as well. Another important observation is that node 1 and nodes 4 to 7 form 180 degree angles, which clearly captures the non-dominance relationship between them in the tree graph. The same is also true for nodes 2 and 3 with nodes 8 to 15.

Random Graph. Figure 7(d) demonstrates the case for a random graph. The graph is generated with repeated Bernoulli sampling of probability 0.467 on a 15 by 15 matrix. The resulting graph contains the same number of edges as that with perfect linear ordering. It is clear that the projections do not follow any pattern and scatter around the origin. However, it is worth noting that node 14 has the highest in-degree, which explains its largest distance to the origin.

5.2.3 Departure from Representative Graphs

For a graph with n nodes, we can calculate the authority score of each node following Equation (20). The network authority information can be described using the vector $\overrightarrow{\Gamma} = (\Gamma_1, \cdots, \Gamma_n)$. Figure 8 shows the bar plots of sorted authority scores for the same examples from Figures 7(a), 7(b), 7(c), and 7(d). The x-axis for each plot shows the indices of nodes and the y-axis shows the authority score. As shown in Figure 8(a), the plot of the linear ordered network resembles the cosine function. This is because the dominance scores are

proportional to the length of arcs between individuals. The plots from Figure 8(b) and 8(c) suggest that the corresponding graphs contain clear tiered hierarchies. The plot of authority scores of a random graph just follows some unknown polynomial, as shown in Figure 8(d).



(a) Perfect linear ordering

(b) Graph with hidden tree structure

(c) Tree graph

(d) Random graph

Figure 8: Authority Score of Each Node from 4 Representative Graphs

Distribution of authority scores from a given network can be compared with those of representative networks such as tree, linear, or random. Based on comparison, we can determine whether a given graph has the same organizational structure of one representative graph. In general, we can conduct tests with the Pareto Q-Q plot [55] to check whether the authority scores of a network follows a particular distribution. A Q-Q plot is used to compare the shapes of distributions, providing a graphical view of how properties such as location, scale, and skewness are similar or different in the two distributions. Moreover, we

can calculate the similarity between the observed graph G with a representative graph G^B as:

$$S(G, G^B) = (\overrightarrow{\Gamma} \cdot \overrightarrow{\Gamma^B})/(\|\overrightarrow{\Gamma}\| * \|\overrightarrow{\Gamma^B}\|). \tag{22}$$

Since the spectral projections are permutation invariant, the calculated scores $\overrightarrow{\Gamma}$ of a given network can be sorted from highest to lowest according to those of the representative scores. As vectors are scaled to unit length in cosine similarity, scalings of vectors by edge weights do not change the results.

For all the above experiments, comparisons with baseline structures such as binary tree or linear ordering are primarily used to find out what type of organizational structure the given network has. If two networks have similar dominance score distributions, they expect to have similar organizational structures. Hence our results can be used to check whether several networks under study are organized similarly. Our method provides a means of determining the similarity of two networks in terms of organizational structure (i.e., comparing their dominance score distributions) as it is often difficult to directly construct synthetic graphs that match a real network with some given organizational structure.

5.2.4 Comparison with PageRank and Similar Methods

The PageRank algorithm [87] is also solely based on the link structure. PageRank interprets a hyperlink from page v to page u as a vote (or endorsement), by page v for page u. A page is important if it is pointed (or endorsed) by other important pages. The PageRank is formally defined as:

$$R(u) = \frac{1-d}{N} + d \sum_{v \in B_u} \frac{R(v)}{N_v}, \tag{23}$$

where u represents a Web page, N is the number of pages, B_u is the set of pages pointing to u, $R(u)$ and $R(v)$ are rank score of page u and v respectively, N_v is the number of out-links from v, and d is a damping factor that is usually set to 0.85. In PageRank, the importance of page u is roughly captured by the sum of the PageRank scores of all pages that point to u. Since a page may point to many other pages, its prestige score should be shared. Hence, in the equation, the rank score of a page v, is evenly divided among its outgoing links.

PageRank outputs a steady-state probability distribution vector where each value represents the possibility that a person would randomly visit the corresponding Web page. In other words, each value indicates the importance rank of the corresponding web page.

PageRank method uses random walk modeled in Markov chain to predict the eigenvector centrality measures. It gives high scores to inflow sinks and marks them as prestige individuals. Our dominance/submissiveness metrics based on skew symmetry measure the asymmetric in/out flow of information based on the given network connection. The pairwise dominance relations and ordering information are considered, so individuals will be segregated into tiers. Those two methods serve different purposes in scoring nodes for a given network. Therefore, PageRank weights more globally from the entire network. while our method weights more on local structures.

It is worth pointing out that many other metrics based on the link structure have been proposed. For example, HITS [54] is an dynamic iterative algorithm based on web links. HITS first expands the list of relevant pages returned by a search engine and then calculates both hub and authority scores simultaneously. Simply speaking, an authority is a page with many in-links, which suggests the page may have authoritative content on some topic and thus many users trust it and thus link to it. A hub is a page with many out-links. A page

with a high hub score often serves as an organizer of the information on a particular topic and points to many good authority pages on the topic. There are two types of measures, centrality and prestige, for measuring node importance in directed networks. Centrality measures such as degree centrality, closeness centrality, and betweenness centrality only exploit the out-links whereas prestige measures such as degree prestige, proximity prestige and rank prestige only utilize the in-links. Our approach based on analysis of asymmetry removes those unnecessary information contained in the symmetric part of the network and purely uses directional information contained in the asymmetric part. Hence, it expects to more accurately capture the network dominance structure. Furthermore, our method combines information from multiple bimensions rather than only the first eigenvector used in PageRank and HITS.

5.3 Empirical Evaluation

The evaluation is performed on a global goods and products trade dataset. The data was collected from United Nations Comtrade Database website. The dataset contains trades among 233 countries and/or trade entities. The trade values were recorded in US dollars for comparison. A trade matrix was generated with the complete export data of year 2014. The reason for choosing 2014 data is that 2015 data was not as complete, since some countries or trade zones did not report their trade volume with each country to the database at the time of collecting the data. As a result, there are 20522 trade records globally after putting the data into a matrix. The density of the resulting matrix is 0.378, which is quite dense. Upon studying the global export or import relationships, we could identify top tier countries that are leading the consumption or production perspective of the global economy.

Depending on the different aspects and relationships that can be extracted or constructed from the dataset, many factors influencing the global trade can be analyzed in detail.

5.3.1 Relationship Network Analysis for Trade Data

The trade network data is converted to skew symmetric form and subjected to SVD. The original trade matrix A contains import information for each column, where A_{ij} indicates the amount of goods and products in US dollar that country i exported to country j in 2014.

Table 6: Top 10 Countries

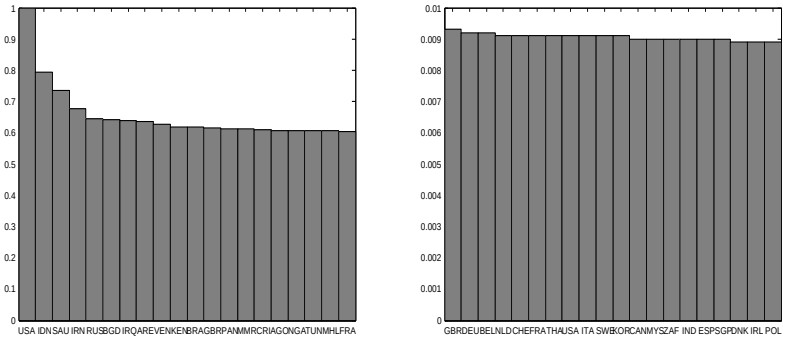
Rank	Authority	PageRank
1	United States	United Kingdom
2	India	Germany
3	Saudi Arabia	Belgium
4	Iran	Netherland
5	Russian Federation	Switzerland
6	Bangladesh	France
7	Iraq	Thailand
8	United Arab Emirates	United States
9	Vietnam	Italy
10	Kenya	Sweden

The cut off parameter for selecting singular values is set to be 0.8. We get a total of 14 singular vectors for calculating the authority scores. The top 10 countries from the authority score and PageRank score results are shown in Table 6. The plots for score results of top 20 countries from our approach and PageRank are also included in Figure 9(a) and Figure 9(b) respectively. The X-axis shows country name labels in ISO country codes. As shown in Figure 9(a), our method shows that USA is the only country that significantly dominates the entire network. It simply indicates that USA is the country that imports a lot more goods and products from around the globe than all the other countries. This result further implies that USA has a very large trade deficit according to the meaning captured by skew symmetry, since the import amount is much bigger than its export amount. This is confirmed from the raw data, where USA has more than 5.5×10^{11} dollars trade deficit in

the goods and products category. We also observe in Figure 9(a) that the other few countries following USA are India, Saudi Arabia, Iran and Russian Federation. These results could indicate that those countries contain the markets that dominate others in driving imports of the global economy.

There could be several factors that would cause a country to rank higher in the import authority rank, but we will only name a few of them. First, countries that are not self sufficient in natural resources need to import them from others. Second, products are imported if a country does not have the complete supply chains required to manufacture them. Third, it is not cost efficient to manufacture the products locally when compared with imported items. There are other important factors such as political issues and trade agreements which will not be discussed here. If we try to categorize the top 10 counties according to the 3 factors mentioned, we will see that United States belong to the first and third. India, Bangladesh, Vietnam and Kenya belong to all three. Saudi Arabia, Iran, Russian Federation, Iraq and United Arab Emirates belong to the second and third. In terms of trade, those countries import products from more partners than those they export to. In terms of supply chains, United States buys products and natural resources; India, Bangladesh and Vietnam manufacture some products but also buy products that the do not make while import resources; Saudi Arabia, Iran, Russian Federation, Iraq and United Arab Emirates are selling resources and buying products. Therefore, authority scores ranked those countries high in the import network.

As observed in Figure 9(b), the PageRank scores on the other hand comprehend the same network quite differently. Due to the nature of this method, it seems edge degrees connected to nodes are more favoured, since the top ranked countries are those with highest degrees.



(a) Dominance scores (b) PageRank scores

Figure 9: Bar Plot of Sorted Dominance Scores vs PageRank Results

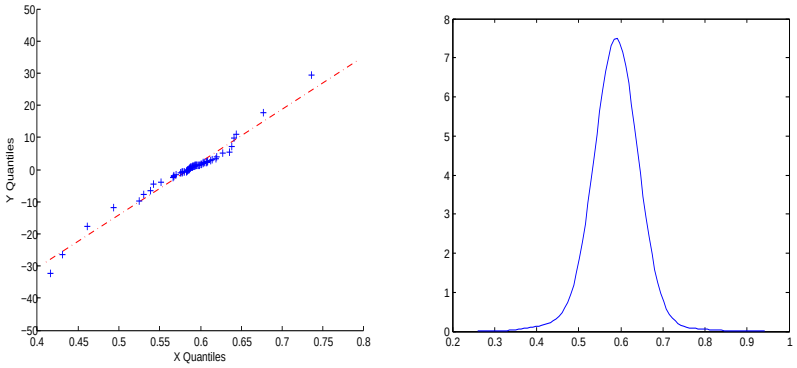
Many countries with similar numbers of edge degrees are ranked the same according to the scores despite the fact that the edges contain drastically different weights. For example, USA has a trade volume almost 3 times bigger than that of Belgium, but they are ranked the same. In reality, it is very anti-intuitive that those two countries are ranked at the same level. However, if we treat the links in the matrix as the preferences of goods and products flow, the random walk based approach such as PageRank could make a little more sense. Nevertheless, we observe that the PageRank scores do not vary much among top 20 countries compared with authority scores. This observation suggests that PageRank may not be an appropriate method to characterize the dominance relationships in terms of goods and products import, but such relationships can be correctly captured by our method.

In this evaluation, the trade data collected only contains products and goods category, but our method could be applied to various perspectives of many other datasets. For example, the surplus or deficit relationships constructed by calculating the absolute difference of import and export amount, the trade data of specific type of goods including agricultural products, crude oil or iron ore, trade data of services could all be possible candidates.

However, due to limited space, we cannot demonstrate our method on each one of those. As a general rule, as long as the relationship matrix can be constructed with a clear meaning from the given network, our method could depict the dominance structure of individuals.

5.3.2 Inference

In this section, an empirical evaluations on import data is conducted to demonstrate how our proposed method can be used to study organizational structures and relationship structures. When certain organizational structure is suspected to exist for a network, we can study the dominance score distribution, which can be used to test against our null hypothesis. When the actual dominance relationship from the raw data is of interest, our method can correctly capture it as long as the the constructed matrix itself represents a clear meaning and contains homogenous links. Furthermore, several networks or the snapshots of the same network at different time can be tested side by side to compare and contrast. A potential application could be the study of the authority scores' distribution changes of trade datasets over multiple years. In all, our method could have many empirical applications in studying the relationship structures of network datasets. The Q-Q plot of import authority scores against Cauchy(0.6, 0.32) distribution is shown in Figure 10(a), while the empirical probability distribution function with kernel smoothing of the scores is estimated as in Figure 10(b). The bandwidth of the smoothing parameter in the kernel is 0.05. At the first glance, the estimated PDF resembles that of a normal distribution, but it has very fat tails and leptokurtosis. Therefore, after cross validations with several other distributions, Cauchy(0.6, 0.32) is our best fit.



(a) Q-Q plot vs Cauchy(0.6, 0.32) (b) Empirical PDF

Figure 10: Q-Q Plot of Trade Dominance and Estimated Empirical Probability Distribution

5.4 Summary

We have conducted dominance analysis based on the canonical analysis of asymmetry and developed the node's authority score metric which combines its dominance and submissiveness. We conducted authority analysis for several representative networks and presented approaches of measuring the departures of a real social network from the representative ones. We emphasize this is first study of the distribution of authority scores of representative graphs. We compare our proposed authority score measure with PageRank score. The empirical evaluation on a global goods and products trade dataset demonstrated the effectiveness of the proposed approach.

In this chapter, we assume the edges in the observed graph are homogenous, i.e., they represent the same type of relationship (flow or email communications) across the entire network. We emphasize many real networks do not satisfy this assumption. For example, the corporate email data contains different types of emails, e.g., announcing meetings, administrative issues, legal issues, and even pure personal matters. The observed links in

the incidence matrices and may not contain the full information about the given dataset, so they may not necessarily indicate the dominance relationships between individuals. How to identify the organizational hierarchies or calculate the authority scores using the observed data could be a challenging problem and worth further investigations. In addition, if streaming data is to be collected, the analysis of the dynamic changes of the dominance structures could also be a topic worth studying.

The preliminary results of this chapter were published in [67].

CHAPTER 6: ANOMALY DETECTION IN DYNAMIC GRAPHS

Identifying vandal users or attackers hidden in dynamic OSN data was shown to be a challenging problem. In this chapter, we develop an automatic spectral-analysis-based attack/anomaly detection approach using a novel combination of the graph spectral features and the restricted Vector Autoregressive (rVAR) model. Our approach utilizes the time series modeling method on the non-randomness metric derived from the graph spectral features to capture the abnormal activities and interactions of individuals.

In this chapter, we demonstrate how to utilize Granger causality test on the fitted rVAR model to identify causal relationships of user activities, which could be further translated to endogenous and/or exogenous influences for each individual's anomaly measures. We also develop an algorithm that could provide causal analysis of the anomaly measures of users from given network data. Case studies of different scenarios are presented to demonstrate the efficacy of the proposed methods and procedures on a labeled WikiSigned dataset.

6.1 Anomaly Detection

Anomalies and outliers refer to data points that behave differently from predefined normal behaviours. Anomaly and outlier detections focus on different aspects of the data, but are related. Some common types of anomalies include point anomalies, contextual anomalies [106], and collective anomalies for sequence data [46], graph data [85] and spatial data [102].

A number of surveys [4,11,13,45,90] have studied extensively on these topics. For applications of anomaly detection, there are cyber-intrusion detection, fraud detection, medical anomaly detection, industrial damage detection, textual anomaly detection, and many others. Some common techniques for detecting anomalies include classification based methods, clustering based methods, nearest neighbor based methods, statistical methods, and spectral analysis methods. Depending on the availability of the labels, anomaly detection techniques can operate as supervised, semi-supervised, and unsupervised approaches.

Node connection pattern based graph anomaly detection was studied early in several works [36, 85, 111]. On the other hand, spectral graph analysis has been shown to have important applications in solving network related problems such as clustering [28, 29, 65, 78, 80, 82, 125], anomaly detection [3, 19, 50, 130, 135], and link prediction [6, 58, 60, 114].

Detecting anomalies in a network under the dynamic setting belongs to sequential anomaly detection, where a detection method tries to find abnormal observations from sequential data. Most OSN streaming data were collected as sequential data, which could form event sequence data, such as system call data [126] or numerical time-series data [18]. User activities are usually complicated in large streaming OSN data. When opposing opinions collide and similar opinions mingle, it is difficult to depict the human to human interactions as simple graphs. As the streaming network data build up, modeling the associated graphs as weighted or even signed graphs could capture the changes of the intensity and the underlying meanings of the user interactions more accurately. In this chapter, we propose a method based on the weighted signed graphs, so that it covers a wider range of research targets.

There have been plenty of works studying the spectral properties of dynamic network

data such as incremental spectral clustering [84], Nystrm low rank approximation [139], and matrix sketching [69], but there are still many open problems in the application of anomaly detection on dynamic graphs. Graph spectral analysis has been shown to be an effective tool for anomaly detection in computer network traffic data. In [105], the authors derived a threshold based on the anomaly metric from the spectral features of the robust Principal Component Analysis (rPCA) for classification. Similarly, the authors in the work [50] proposed a threshold based on the anomaly metric derived from the principal eigenpairs of the associated adjacency matrix to classify individuals. In another work [112], the authors proposed using compact matrix decomposition (CMD) to compute the sparse low rank approximations of the adjacency matrix. The approximation error of CMD and the observed matrix was used to quantify the anomaly.

Although the works [50, 105, 112] used spectral analysis based methods for anomaly detection on time series data, there are two major shortcomings. First, instead of using statistical modeling approach to analyze the underlying structural correlations of the time series data systematically, each of the works only derived a threshold to evaluate the data points at each time frame individually. In many scenarios, categorizing data into different contexts and analyzing whether a particular piece of data is anomalous under its associated context is not straightforward. The other drawback is that, the methods proposed in those works lacked the ability to analyze the endogenous and/or exogenous causes for the observed anomalies. Since most relationship graphs are generated from the interaction information of the streaming OSN data, such interactions could cause the observed anomaly metrics to be correlated thus more complicated than network traffic data. Therefore, both endogenous and exogenous influences in the observed time series data need to be analyzed

simultaneously so that the underlying casual relationships could be identified.

There exist extensive literatures on the applications of time series analysis methods such as Autoregressive (AR) model [32], Autoregressive Moving Average (ARMA) model [2, 120], VAR model [70] and Vector Autoregressive Integrated Moving Average (VARIMA) model [16, 121] in the research of outlier detection. However, their applications in anomaly detection in streaming OSN data have been relatively scarce. In this chapter, we propose to use the rVAR model to simulate the interactions and correlations of the observed anomaly measures of nodes, since it is a relatively simple multivariate time series analysis technique that could be used to evaluate correlated variables simultaneously. In addition, the fitted model could serve as the input for the subsequent casuality analysis process.

Casuality analysis methods such as Granger causality [42] and conditional Granger causality index (CGCI) [38] for multivariate time series models were proven to be very useful for identifying casual relationships amongst variables. Therefore, we adopt them to analyze the fitted rVAR model and identify both endogenous and exogenous influences in the observed anomaly measures for each node.

To summarize this chapter, we incorporate the dynamic spectral features from the steaming network data with the rVAR model to develop an automatic fraud/attack analysis method. As for the explanatory and response variables, we propose to use the modified anomaly metric based on the node non-randomness measure derived from the adjacency spectral coordinates from the works [134] and [135], which could quantify how randomly nodes link to each other. We then propose to use the Granger causality analysis to identify the causal relationships amongst individuals. We also develop an algorithm based on the proposed anomaly analysis procedures. Furthermore, several case studies on a partial Wik-

iSigned dataset are conducted to demonstrate how the Granger causality analysis could be used to interpret the fitted rVAR model.

Our contributions are as follows.

- We propose to use a statistical modeling approach (rVAR model) to analyze the structural correlations of the node anomaly measures.
- We propose to use Granger causality to identify the endogenous and/or exogenous influences of node anomalies.
- We derive an algorithm for anomaly analysis from streaming OSN data.
- We present case studies using the proposed algorithm on a real dataset.

6.2 Background Information

We model a dynamic network dataset as a sequence of graphs along the time dimension as G_t, where $t = 1, \cdots, T$. Each graph could be viewed as a snapshot of the network at time t. Hence, if we treat each snapshot at time t as a perturbation from the previous time $t-1$, the associated adjacency matrix can be written as $A_t = A_{t-1} + E_t$, where E_t contains the changes between two adjacent snapshots of G_{t-1} and G_t. There are three challenges involved in identifying dynamic attacks. The first challenge is to identify the correct snapshot time windows when the suspicious activities occur. The second challenge is to distinguish anomalies due to attacks and significant changes due to regular user activities. The third challenge is to identify the endogenous and/or endogenous sources of the causes for the anomalies. Therefore, the task for detecting anomalies could be achieved by addressing the above challenges.

Dynamic networks focus on cognitive and social processes of users and can model the addition and removal of relations and interactions in OSNs. The dynamic changes of user activities are assumed to follow some particular probabilistic model such as the random walk or preferential attachment. When the perturbation E_t contains changes that deviate from the expected statistics under the assumed probabilistic model of normal behaviors, such events could be captured and treated as suspicious. The rVAR model could be used to analyze the underlying correlations amongst individual's anomaly measures and make subsequent casuality inferences.

6.2.1 Graph Spectral Projections

The eigenvalues $(\lambda_1, \cdots, \lambda_n)$ of a given adjacency matrix A are assumed to be in descending order when real. The corresponding eigenvectors $(\boldsymbol{v}_1, \cdots, \boldsymbol{v}_n)$ are sorted accordingly. The spectral decomposition of A takes the form $A = \sum_i \lambda_i \boldsymbol{v}_i \boldsymbol{v}_i'$. Many survey works [8, 24, 71, 98, 101] stated that the algebraic properties of the adjacency matrix are closely related to the underlying graph connectivity. Therefore, when the nodes are projected into the associated spectral space spanned by the chosen eigenvectors, such properties could be used to analyze the graph structure related problems. In this chapter, we use the adjacency eigenspace for the spectral projections as shown in the works [132] and [65].

$$
\begin{array}{c}
\begin{array}{cccccccc}
\boldsymbol{v}_1 & & \boldsymbol{v}_i & & \boldsymbol{v}_K & & \boldsymbol{v}_n
\end{array} \\
\downarrow \\
\boldsymbol{\alpha}_u \rightarrow \left(\begin{array}{c|c|c|c}
\boldsymbol{v}_{11} \cdots & \boldsymbol{v}_{i1} & \cdots \boldsymbol{v}_{K1} \cdots & \boldsymbol{v}_{n1} \\
\vdots & \vdots & \vdots & \vdots \\
\hline
\boldsymbol{v}_{1u} \cdots & \boldsymbol{v}_{iu} & \cdots \boldsymbol{v}_{Ku} \cdots & \boldsymbol{v}_{nu} \\
\hline
\vdots & \vdots & \vdots & \vdots \\
\boldsymbol{v}_{1n} \cdots & \boldsymbol{v}_{in} & \cdots \boldsymbol{v}_{Kn} \cdots & \boldsymbol{v}_{nn}
\end{array}\right)
\end{array}
\tag{24}
$$

For a given network, the spectral coordinates of nodes derived from the adjacency eigenspace are illustrated in Equation (24) (To avoid confusion and abuse of notation, the spectral coordinates are introduced again here with different notations). If we assume that the eigenvectors are sorted so that the first K eigenvectors correspond to the ones that form the perturbed Perron Frobenius simple invariant subspace as in work [65], the row vector $\boldsymbol{\alpha}_u = (\boldsymbol{v}_{1u}, \boldsymbol{v}_{2u}, \cdots, \boldsymbol{v}_{Ku})$ is the spectral coordinates used for the projection of node u.

There are other spectral projection methods such as Laplacian, Normalized Laplacian, or SVD that use factorized adjacency matrices. However, it would be easier to derive suspiciousness metric from the adjacency spectral coordinates, since they are not adjusted or balanced specifically for clustering and segmentation purposes as in the other methods.

6.2.2 Non-randomness Measure

We choose the node non-randomness metric from the work [134] as the input variables in the VAR model. The node non-randomness is derived from the spectral coordinates to quantify how random a node is in terms of its connections. The measure was shown to identify random link attacks in the static spectral space in the work [135]. In this chapter,

we adapt the measure in the dynamic spectral space and use it to identify anomalies such as random link attacks in streaming OSN data. The edge and node non-randomness measures are defined as:

1. The edge non-randomness $R(w, u)$:

$$R(w,u) = \sum_{i=1}^{K} \boldsymbol{v}_{iw}\boldsymbol{v}_{iu} = \boldsymbol{\alpha}_w\boldsymbol{\alpha}'_u = \|\boldsymbol{\alpha}_w\|_2\|\boldsymbol{\alpha}_u\|_2 \cos(\boldsymbol{\alpha}_w, \boldsymbol{\alpha}_u), \tag{25}$$

   where v_{iw} is the w-th entries of the i-th chosen eigenvector, up to K eigenvectors.

2. The node non-randomness $R(w)$:

$$R_w = \sum_{u\in\Gamma(w)} R(w,u), \tag{26}$$

   where $\Gamma(w)$ denotes the set of neighbor nodes of w.

This metric was shown to be effective in identifying random link attacks in static simple graphs. For directed signed graphs, according to the theoretical results from Theorem 3, Theorem 5 and the error bound analysis in [129], the edge nonrandomness metric could remain the same, but with an approximated error term $O(\frac{E_t}{\lambda_i}v_i)$. When the inter cluster edges (must satisfy the assumptions of the theorems) are treated as perturbations, the changes in Perron Frobenius eigenvectors corresponding to clusters are bounded, so the error term associated with the edge nonrandomness metric should be bounded as well. Therefore, we could conclude that the metric works for directed graphs in general. However, for weighted graphs, the node nonrandomness metric needs to be further adjusted. In this chapter, we will modify this metric in subsection 6.3.2.

6.2.3 Vector Autoregression

Vector Autoregressive model is a time series analysis approach for analyzing multivariate data. It tires to capture the changes and interferences of multiple variables over time, where each variable is explained by the lagged values of itself and those of other variables. The following Equation (27) shows the general form of a n-variable VAR model with lag p:

$$\begin{pmatrix} x_{1,t} \\ \vdots \\ x_{n,t} \end{pmatrix} = \begin{pmatrix} c_1 \\ \vdots \\ c_n \end{pmatrix} + \sum_{i=1}^{p} \begin{pmatrix} \beta_{11,i} & \cdots & \beta_{1n,i} \\ \vdots & \ddots & \vdots \\ \beta_{n1,i} & \cdots & \beta_{nn,i} \end{pmatrix} \begin{pmatrix} x_{1,t-i} \\ \vdots \\ x_{n,t-i} \end{pmatrix} + \begin{pmatrix} \varepsilon_{1,t} \\ \vdots \\ \varepsilon_{n,t} \end{pmatrix}. \tag{27}$$

It can be written in a vector form as:

$$X_t = \boldsymbol{c} + \sum_{i=1}^{p} \boldsymbol{\beta}_i X_{t-i} + \boldsymbol{\varepsilon}_t \equiv \Pi' Z_t + \boldsymbol{\varepsilon}_t, \tag{28}$$

where $X_t = (x_{1,t}, \cdots, x_{n,t})'$, $\boldsymbol{\varepsilon}_t = (\varepsilon_{1,t}, \cdots, \varepsilon_{n,t})'$, $\boldsymbol{c} = (c_1, \cdots, c_n)'$ is the vector of constants, $\boldsymbol{\beta}_i$s are the matrices of parameters as shown in Equation (27), $Z_t' = (\mathbf{1}_{n\times 1}, X_{t-1}', \cdots, X_{t-p}')$, and $\Pi' = (\boldsymbol{c}, \boldsymbol{\beta}_1, \cdots, \boldsymbol{\beta}_p)$.

The maximum likelihood estimate (MLE) of $\boldsymbol{\alpha}_i$ has a reduced form as:

$$\hat{\boldsymbol{\beta}}_i = \hat{\Pi}' G_i, \tag{29}$$

where $\hat{\Pi}' = (\sum_{t=p+1}^{T} X_t Z_t')(\sum_{t=p+1}^{T} Z_t Z_t')^{-1}$, and G_i is a $(np+1) \times n$ matrix with value 1 from row $(i-1)n+2$ to row $in+1$ and value 0 in other rows.

The parameters of the model could be estimated using the multivariate least squares (MLS) approach, which is the multivariate version of the ordinary least squares (OLS) method. Since each individual model is evaluated using the same set of explanatory vari-

ables, MLS estimator is equivalent to applying the OLS method to each model separately. When the error terms are assumed to be multivariate normally distributed, the MLS estimator is equivalent to the conditional maximum likelihood estimator (MLE) as shown in Equation (29). The OLS for the i-th model seeks to minimize the sum of squared errors (SSE), $\sum_{t=p+1}^{T} \hat{\varepsilon}_{i,t}^2$, with the objective function:

$$\arg\min_{\boldsymbol{\alpha}}(SSE_i) = \arg\min_{\boldsymbol{\alpha}}(\hat{\boldsymbol{\varepsilon}}_i'\hat{\boldsymbol{\varepsilon}}_i), \tag{30}$$

where $\hat{\boldsymbol{\varepsilon}}_i = (\hat{\varepsilon}_{i,p+1}, \cdots, \hat{\varepsilon}_{i,T})'$ for any $i = 1, \cdots, n$.

Upon solving the objective function for each model and consolidating the results into the vector form, we get the reduced form for the estimators as in Equation (29).

In order for the estimators to be accurate and the subsequent statistical inferences to be reliable, the following assumptions are generally made for the VAR model:

- **Linearity**: The regression model is linear in parameters.
- **Exogeneity**: The conditional mean of residuals given the information of explanatory variables is zero, $E(\varepsilon_i|X_1, \cdots, X_n) = 0$.
- **Homoscedasticity**: The conditional variance of residuals given the information of explanatory variables is a constant, $var(\varepsilon_i|X_1, \cdots, X_n) = \sigma_i$.
- **No multicollinearity**: The matrix of explanatory variables has full column rank.
- **Normality**: The conditional distribution of residuals given the information of explanatory variables is normal.

In order for the estimators of the VAR model parameters to exist, it is required that $np <$

T, where n is the number of variables, p is the lag chosen, and T is the observation length. In the ideal condition, np should be much smaller than T so that the estimations could be reliable. In applications, time series data could be nonstationary, but for multivariate time series analysis, we could still obtain correct regression results as long as the series entering the model are cointegrated. This issue is handled in section 6.4. For large and complex streaming network data, all the above assumptions may not strictly hold. Carefully reformatting and cleaning the streaming data could partially solve the problems, but those methods are not our focus, so we assume those conditions to hold true.

6.3 Methodology

6.3.1 Overview

Depending on the content of the given streaming OSN data, the relationship network for individuals could be changing constantly. The first step of our anomaly detection method is to build network snapshots. As shown in Figure 11, depending on the data content, edges could be built directly or indirectly such as interpreted from individuals' actions towards some common objects. As more data are being streamed in, the network graph will change accordingly. As a result, the associated adjacency matrices could represent signed and weighted graphs as shown in the last row. In this example, the edge weight between nodes u and v changes from -1 through 0 to 1. Therefore, the resulting network could be signed and weighted.

After obtaining the adjacency matrices of the network snapshots, we can construct time series of the anomaly metric for each node at each time. In this section, we modify the nonrandomness measure and use this new metric to form the time series data from network

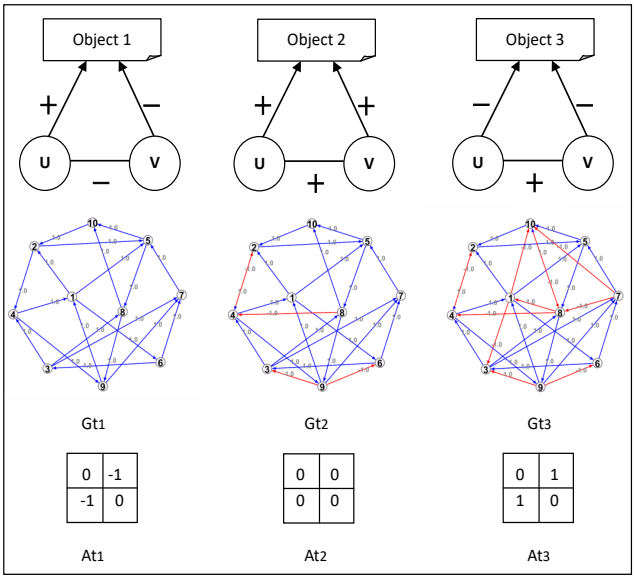


Figure 11: Graph snapshots can be built from the streaming OSN data.

snapshots. Such time series data could allow us to explore the correlation and causality of node activities along the time dimension using multivariate time series analysis techniques such as the VAR model. The fitted model could then be used for causality analysis of the interactions of node behaviors.

6.3.2 Adjusted Node Nonrandomness Measure

For signed and weighted graphs, the node nonrandomness measure in section 6.2.2 may no longer be accurate, since the degree of a node can exceed n or be negative. In order for the measure to work, we propose the following adjusted node nonrandomness:

Result 1. *Let w be a node and $\Gamma(w)$ be its neighbors from a signed graph. The adjusted node nonrandomness measure is*

$$\check{R}_w = \frac{\sum_{u\in\Gamma(w)} R(w,u)}{\sum_{v\neq w} 1_{[A_{w,v}\neq 0]}}, \tag{31}$$

where $R(w,u)$ is the edge nonrandomness and A is the adjacency matrix.

This modification normalizes the node nonrandomness metric by its number of connections, since the edge nonrandomness metric holds true for directed signed and weighted graphs with an error term. If the node nonrandomness metric is normalized as shown above, the error term will shrink as well. As a result, the new metric could better approximate the true metric than the old metric would do. However, the approximation error still exists. In order to construct a more sophisticated node nonrandomness measure for signed and weighted graphs, works based on the probabilistic modeling of the graph edge weight distributions are needed, which is out of the scope of our work.

Under the dynamic OSN setting, the past behaviours of nodes and their correlated ones could be incorporated in a rVAR model, so the influences of suspicious activities such as random link attacks could be studied through multiple snapshots of the network to provide an analysis over the time dimension. For a given node w, it will have a sequence of observed node nonrandomness measures $(\check{R}_{w,1}, \cdots, \check{R}_{w,T})$ based on network snapshots. The observed values could change according to how the node and its neighbors act. By fitting the time series of any selected set of nodes into an rVAR model, we can identify the causal and dependency relationships amongst individuals' suspiciousness measures.

Due to the possible existence of various types of anomalies for a given dataset, it would be better to derive different types of metrics as explanatory variable for different scenarios such as coordinated attacks focusing one group, targeted attacks focusing on specific nodes, synchronized attacks on random groups and more. Such an approach could make the statistical inferences on the casual relationships much cleaner, since each type of explanatory variables can be used to analyze some specific aspect of the network properties or detect a specific type of anomaly.

6.3.3 Event Based Time Series

For large streaming OSN data, individual activities tend to be sparse, so the time series data formed by nonrandomness measures could have activities of mixed frequencies. If network snapshots were taken at time intervals of some fixed length, we may miss activities of nodes that are only active for a short period of time or very sparsely. As a result, the assumptions made earlier for the VAR model may not be met strictly. Therefore, how to construct reliable time series sequences from the steaming data becomes a critical issue.

The network activities captured at different time could be modeled by matrix perturbations, where $A_t = A_{t-1} + E_t$. If we take snapshots at each time when activities (captured by E_t) are observed in the network, then an event based time series data could be constructed. This format could capture all the activities of the network and transform them into a uniformly spaced observations along the adjusted time dimension. As a result, the reformatted time series data could address the mixed frequency problem for observed data, which could cause the VAR model to be fitted incorrectly. There are other approaches to handle such an issue, but the event based snapshots approach is very easy to implement and it could capture all the activities happened in the network. Therefore, it is chosen to format the observed node anomaly metric data captured at each event. It is possible that several individuals are active in the same time, so all such activities are recorded into a single E_t. If we construct the network snapshots accordingly and calculate the node anomaly measures $(\check{R}_{w,1}, \cdots, \check{R}_{w,T})$ for each node, the calculation results will become a set of event based time series data.

6.3.4 Variable and Model Selection

Due to the large sizes and long time spans of streaming OSN data, the observed time series data tend to cause the VAR model to have a large number of explanatory variables. Such datasets could cause the model to overfit. Hence, it is necessary for us to utilize some variable and model selection methods to obtain reliable and efficient estimation results.

In this subsection, we will explore the rVAR model and the Least Absolute Shrinkage and Selection Operator (LASSO). rVAR uses prior knowledge to regulate the parameters. LASSO uses information criterion as the model fitting quality control measure. Once the parameters are identified as statistically significant to the model, the parameters with greater absolute values indicate that the corresponding explanatory variables explain more of the response variable.

After the model fitting process, the features of the fitted model could be extracted for the classification purpose. In addition, causality analysis could also be conducted to reveal the causal relationships from the variables.

LASSO [116] is a special case of the Least Angle Regression (LAR) [30]. It uses penalized regression techniques to control the total number of nonzero parameters entering a regression model. Since the estimations of any VAR model could be treated as a sequence of OLS on each variable when the residuals are assumed to be multivariate normally distributed, the model for each individual (node) could be adjusted one by one using LASSO. This procedure is the same as VAR-LASSO [47]. The objective function for LASSO in Equation (32) is a penalized version of Equation (30) such that

$$\arg\min_{\alpha\in\mathbb{R}^p}\{\frac{1}{N}\|X_t-\Pi' Z_t\|_2^2+\lambda\|\beta\|_1\}, \tag{32}$$

where λ needs to be cross validated. The quality of the LASSO result could be inferred from C_p score, which is a validation process based on Akaike Information Criterion (AIC) [5], where $C_p = -2L + 2(n + 2np)$ with L as the log-likelihood values. The best fitted model should have the lowest C_p score which can be negative for negative valued data. LASSO could not perform well when there are a large amount of candidate variables, since it is time consuming to use cross validation procedures to find the optimal value for λ.

The rVAR method could take a binary restriction matrix $\dot{r}$ and remove variables corresponding to the zero locations. The vector form of the rVAR(P) model is:

$$X_t = \Pi'(\dot{r} \odot Z_t) + \boldsymbol{\varepsilon}_t, \tag{33}$$

where $\dot{r}' = (\mathbf{1}_{n\times 1}, r_1', \cdots, r_P')$.

When analyzing the network data, the restriction matrix could be the concatenation of any matrix representing the desired node connectivity such as 1-step or 2-step neighbor connectivity matrix at a specific lag. In applications, A_t, which is the most recent observed adjacency matrix for a given rVAR model, could be used as r_ps for $p \in (1, \cdots, P)$. The only drawback is that some previously unconnected nodes could be included in the model for some certain lags. However, as long as the number of added variables are small, those extra variables would not influence the model too much. Therefore, during the rVAR model estimation process, only the variables representing connected nodes could have nonzero parameters. As a result, the restricted model based on the network connectivity could

greatly reduce the ambiguities caused by correlated variables representing disconnected nodes, reduce the number of variables entering the model and reduce the risk of having rank deficient data.

The most important reasons why this model is a better choice are that we have the complete prior knowledge of the network connectivity at any previous time and that disconnected nodes should not have any casual relationships among them. The first reason guarantees the existence of all the A_ts, while the second can assure that the chosen variables entering the rVAR model would make sense in terms of the associated network structures. Furthermore, we could avoid the uncertainties such as computational complexities and finding the optimal λ induced by using LASSO like methods when choosing variables. Therefore, in this chapter, we propose to build a rVAR model for each node with its close neighbors, since it would also be beneficial for the causality analysis.

6.3.5 Causal Analysis with Granger Causality

After fitting the rVAR model on each individual and its neighbors, the dependencies and casual relationships of their anomaly measures could be analyzed. The concept of causality analysis for time series data was introduced by Wiener [127] and later formulated by Granger [42]. The classical Granger causality test is an F test to validate if by adding an extra explanatory variable could better explain the current response variable. That is, for models:

$$\text{Model 1:} \quad y_t = \alpha y_{t-1} + \varepsilon_t \tag{34}$$

$$\text{Model 2:} \quad y_t = \alpha y_{t-1} + \beta x_{t-1} + \varepsilon_t, \tag{35}$$

the hypothesis $H_0 : \beta = 0$ and $H_1 : \beta \neq 0$, are tested against each other. Then, the F-statistics

$$F = \frac{(RSS_1 - RSS_2)/(p_2 - p_1)}{RSS_2/(T - 1 - p_2)} \sim \mathcal{F}(p_2 - p_1, T - 1 - p_2),$$

where RSS_i and p_i are the residual sum of squares and the number of parameters of model i respectively, has a F-distribution with $(p_2 - p_1, T - 1 - p_2)$ degrees of freedom if the null hypothesis holds.

When H_1 holds, it simply suggests that X_{t-1} "Granger causes" Y_t, which means that it helps forecast Y_t, but it does not conclude that X_{t-1} causes Y_t. Under such conditions, a completely unrelated variable X may help forecast Y even if it does not cause/relate to Y. In this case, we have $E(Y_t|Y_{t-1}, X_{t-1}) \neq E(Y_t|Y_{t-1})$, where X_{t-1} could help explain Y_t in the model. Such a result could cause confusions when analyzing the fitted rVAR model for any large streaming data. Due to the possible nonstationarity of time series data mentioned in subsection 6.2.3, many variables could Granger cause others even if the corresponding nodes are not connected (directly or indirectly) in the associated relationship network. However, with the help of rVAR model, such risks are greatly reduced, since nodes that are not connected are excluded from the model.

Two adaptations of Granger causality test for multivariate regressions are stepwise forward selection and stepwise backward elimination of the explanatory variables. In both cases, each variable's lagged terms are tested one by one using the models in Equations (34) and (35). Both methods will use np tests in total, which are time consuming but can provide more specific casuality analysis for each individual at each lag.

Another adaptation for multivariate regressions such as the rVAR model called condi-

tional Granger causality index was proposed in the work [38]. It can be used to analyze the dependencies of multiple time series. For models:

$$\text{Model 1: } X_{i,t} = \sum_{k \neq j}^{n} (\beta_{ik,1} X_{k,t-1} + \cdots + \beta_{ik,p} X_{k,t-p}) + \varepsilon_{i,t} \tag{36}$$

$$\text{Model 2: } X_{i,t} = \sum_{k=1}^{n} (\beta_{ik,1} X_{k,t-1} + \cdots + \beta_{ik,p} X_{k,t-p}) + \varepsilon_{i,t}, \tag{37}$$

the residual variances $\hat{\sigma}_1^2$ of model 1 and $\hat{\sigma}_2^2$ of model 2 are compared to quantify the causal effect from X_j to X_i as

$$CGCI_{X_j \to X_i} = \ln \frac{\hat{\sigma}_1^2}{\hat{\sigma}_2^2}. \tag{38}$$

The index value is close to zero when X_j does not Granger cause X_i. Its statistical significance is evaluated using the following F-statistics

$$F_{CGCI} = \frac{(RSS_1 - RSS_2)/p}{RSS_2/(T - p - p_2)} \sim \mathcal{F}(p, T - p - p_2),$$

where p is the number of lags, T is the observation length, and p_2 is the number of parameters in model 2. The casual effects based on all the past lags of a selected explanatory variable $X_{j,t-1}, \cdots, X_{j,t-p}$ are tested to see whether they Granger cause the response variable.

Based on the causality analysis results, the sources of endogenous and exogenous causes for each node's anomaly measures could be identified. Therefore, we can distinguish whether the node itself is anomalous or it is caused by adjacent neighbors's behaviours. For spectral graph analyses on streaming network data, since nodes' spectral coordinates are constantly influenced by their neighbors' activities, it is crucial for us to correctly identify the source causes affecting their anomaly measures. Several case studies are conducted

in section 6.5 to demonstrate how such analyses could be used to reveal the causal relationships of node anomaly measures.

6.3.6 Completeness of Conditional Information

As mentioned in the previous two subsections, a rVAR model needs to be built for each target node with its close neighbors before the casuality analysis could be performed. The main reason for this choice is that we need to have complete conditional information for building the multivariate model for the target node.

If the target node w has n_1 1-step neighbors, then the resulting rVAR(p) model will produce $n_1 + 1$ multivariate AR models, one for each node. Let $\Gamma(w)$ denote the set of the 1-step neighbors of w. The fitted restricted multivariate AR model $rVAR(p)_w$ of node w is equivalent as the conditional expectation $E(w|\Gamma(w))$. By looking at the model $rVAR(p)_u$ of any node u from $\Gamma(w)$, unless all of the nodes from $\Gamma(u)$ are also in $\Gamma(w)$, $\Gamma(w)$ could not provide complete conditional information for node u as $\Gamma(u)$ does. The direct result from this scenario is that the model $rVAR(p)_u$ is not based on the complete conditional information. Any subsequent statistical inferences such as casuality analysis and variable selections may be inaccurate.

By building the rVAR(p) model for each target node with its directly connected neighbors, we are able to capture all the influences from exogenous sources for it. This complete information set of nodes is analogous to the Markov blanket concept such that

$$Pr(w|\partial w, u) = Pr(w|\partial w), \tag{39}$$

where w and u are distinct nodes in a Bayesian network, and ∂w is the Markove blanket

of w. ∂w is the set containing node w's parent nodes, child nodes, and its children's other parent nodes. Although nodes in a Bayesian network are connected by directed links, ∂w contains the complete knowledge for predicting the behavior of w. All other nodes that are not in ∂w are conditionally independent with node w when given ∂w.

For general networks, graph perturbation theories tell us that the node spectral projections could be perturbed by activities of neighbors m-step away. Therefore, it is still possible for the rVAR model to capture the exogenous influences from k-step neighbors. However, all the influences need to be passed to the target node form its 1-step neighbors and we need have a cutoff point to limit the number of parameters entering the rVAR model. Hence, the 1-step neighbors are used as the default choice in our experiments. In case where additional studies of the multi-step neighbors are of interest, the model could be extended accordingly. The case studies in Section 6.5 could serve as examples on how to deal with those two cases.

6.4 Algorithm

In this subsection, we propose our algorithm for applying rVAR method on streaming OSN data for anomaly detection in two functions shown in Algorithm 3 and Algorithm 4.

The algorithm takes threes steps to complete the task. Firstly, the node nonrandomness measures are calculated from the spectral coordinates at each snapshot of the network. Secondly, for each target node, the chosen neighbors are incorporated to fit the rVAR(p) model. Johansen cointegration test from the work [52] is also performed at this step to prevent spurious regression in case where the associated time series data are integrated. Lastly, either stepwise backward elimination Granger causality analysis or CGCI method

Algorithm 3 $OSN\_rVAR\_Granger$: Anomaly analysis of the streaming OSN data using stepwise backward elimination Ganger casuality/CGCI on the rVAR model of node nonrandomness measures

Input: $(A_1, \cdots, A_T), n, T, K, p, m, Arg$

Output: B, B_Ind

I/O: The inputs are the adjacency matrices $(A_1, \cdots, A_T)$, size of the users n, observation length T, number of the eigenpairs K, lag p, number of the steps of neighbors m, causality analysis method Arg. The outputs are parameters for fitted rVAR models B, and causality indicators B_Ind

1: **for** t from 1 to T **do**
2: Compute eigenvectors $(v_1, \cdots, v_K)$ of A_t corresponding to the largest K eigenvalues $(\lambda_1, \cdots, \lambda_K)$;
3: **for** w from 1 to n **do**
4: Calculate the node nonrandomness score normalized by its number of connections $\check{R}_{w,t} = \frac{R_{w,t}}{\sum_{u \neq w} 1_{[A_{wu,t} \neq 0]}}$;
5: **end for**
6: **end for**
7: **for** w from 1 to n **do**
8: $S \leftarrow w \cup \Gamma(w)_m$;
9: **for** u from the $m-$step neighbor set $\Gamma(w)_m$ **do**
10: Perform Johansen cointegration test on the time series $\check{R}_{w,\cdot}$ and $\check{R}_{u,\cdot}$;
11: **if** not cointegrated **then**
12: $S \leftarrow S \setminus u$;
13: **end if**
14: **end for**
15: Fit rVAR(p) model on the restricted set of nodes S with their corresponding time series $\check{R}_{s,\cdot}$, where $s \in S$;
16: Extract $B_w \leftarrow (\beta'_{w,1}, \cdots, \beta'_{w,p})$;
17: $B\_Ind_w \leftarrow Granger\_Causality(B_w, \check{R}, S, Arg)$;
18: **end for**
19: Return B;
20: Return B_Ind;

could be used to perform Granger casuality analysis of the node nonrandomness time series. For Algorithm 3, in lines 1-6, we calculate the node nonrandomness for each node at each network snapshot. In lines 9-14, we remove the observations that are not cointegrated with the target node. In lines 7-18, we fit rVAR model for each node in line 16 and evaluate its Granger causality in line 17, where Algorithm 4 is called. For Algorithm 4, depending on the Granger causality method used, either stepwise backward elimination Granger causality

analysis is used in lines 1-10, or CGCI method is used in lines 11-20. The significance of the calculated F-statistic is determined by looking up the F-statistic table, where it is common to choose 0.05 alpha level.

Algorithm 4 $Granger\_Causality$: Granger causality analysis for the multivariate AR model of a given node w

Input: $B_w, \check{R}, S, Arg$
Output: B_Ind_w

1: **if** Arg==“Stepwise backward elimination” **then**
2: $B\_Ind \leftarrow 0$
3: **for** Each $\beta \in B_w$ **do**
4: Get P_β, the p-value for the F-statistic from the Granger causality test;
5: **if** β is significant **then**
6: $B\_Ind_{w,\beta} = 1$;
7: **end if**
8: **end for**
9: Return B_Ind_w;
10: **end if**
11: **if** Arg==“CGCI” **then**
12: $B\_Ind \leftarrow 0$
13: **for** Each $s \in S$ **do**
14: Get $P_{CGCI_{s\rightarrow w}}$, the p-value for the F-statistic from the CGCI;
15: **if** $CGCI_{s\rightarrow w}$ is significant **then**
16: $B\_Ind_{w,s} = 1$;
17: **end if**
18: **end for**
19: Return B_Ind_w;
20: **end if**

CGCI can provide a faster analysis by checking all the lags of a given variable together, but stepwise backward elimination Granger analysis could provide the most detailed analysis for each variable at each lag. The final outputs of the algorithm are two cell arrays, B which contains the parameters for the rVAR model of all nodes and B_Ind which contains the corresponding indicators based on the causality analysis. Further inferences could be conducted based on the causality analysis results. In the next section, we demonstrate our

algorithm on a real dataset and visualize some of the results.

6.5 Empirical Evaluation

In this section, we conduct evaluations using a partial UMDWikipedia dataset from [56]. The original dataset contains 770,040 edits of Wikipedia pages made by both vandal and benign users between January 01, 2013 and July 31, 2014. Since user edits were recorded with precision to seconds, we convert the time stamp into Matlab date numbers which starts from January 00, 0000 for easier arrangement of the time series data. We use the Black list DB and White list DB files which contain 17,027 vandal users and 160,651 edits and 16,549 benign users with 609,389 edits respectively. For our study, we keep only edits on the "Article" page type. Since we focus on analyzing the dynamic interactions of the user behaviors, only pages edited by more than 3 unique users and users editing more than 3 unique pages are kept. After cleaning the data, we have 17,733 edits and 805 users spanning over 10,451 unique event times, where there are 456 benign users and 349 vandal users. The associated relationship network is interpreted using the same method as the example presented in Figure 11. This is a labeled dataset that can form a dynamic signed and weighted network representing user interactions.

In this section, we use this partial dataset to investigate several case studies to demonstrate how Granger causality analysis can help us identify the causes for the observed anomaly measures. We use the rVAR(5), which is the rVAR model of lag 5, on 1-step neighbors for all the case study examples unless further specified. More lags could better capture the causality influences, but more computational time is needed. For the case studies, we use 5 time events as the interval to build the time series data, so we have 2,091 time

frames.

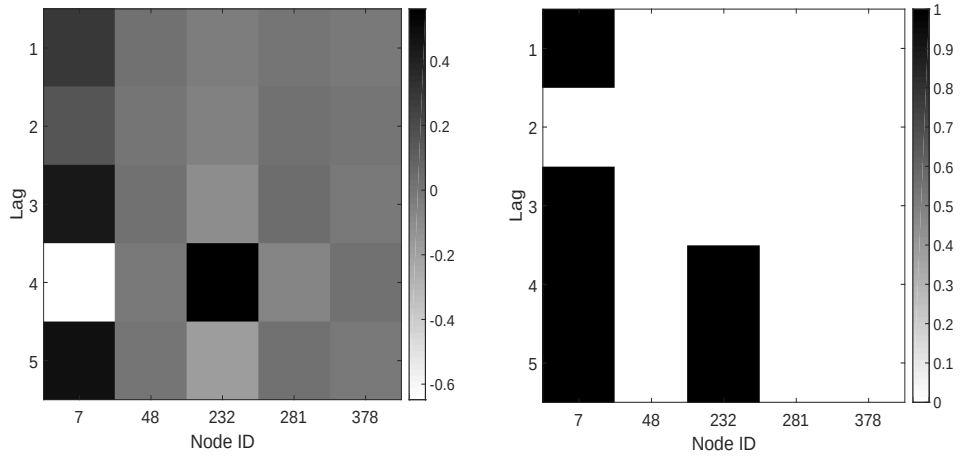
6.5.1 Case Study 1

The target node is 7 and its 1-step neighbors are 48, 232, 281 and 378. The node anomaly measure variables are relabeled as X_1 to X_5 respectively. The adjusted model using step-wise backward elimination multivariate Granger causality analysis method is

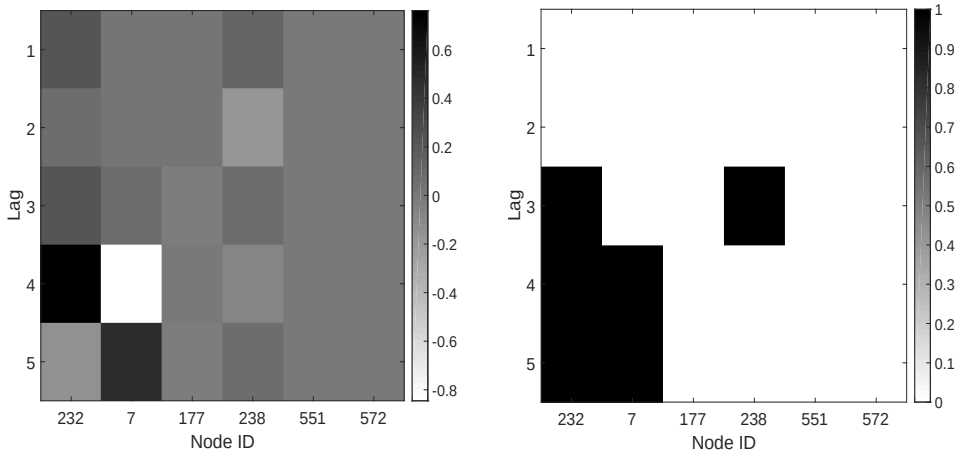
$$\begin{aligned} X_{1,t} = {} & 0.288X_{1,t-1} \\ & + 0.443X_{1,t-3} \\ & - 0.648X_{1,t-4} + 0.56X_{3,t-4} \\ & + 0.467X_{1,t-5} - 0.186X_{3,t-5}, \end{aligned} \tag{40}$$

where $c = 0.00007$ is not significant with t-statistic value of 1.1679. The parameter vector and the associated significance indicators from Granger casuality results are shown in Figure 12(a) and Figure 12(b). For the parameter vector figures in this section, each row represents the parameters for all the variables of a certain lag and each column represents the parameters of all the lags for a certain variable. For the causality indicators figures, each shaded location suggests that the corresponding column variable Granger causes the row variable.

The Granger causality significance indicator grid based on the F-statistics is shown in Figure 12(b). It suggests that the lag 4 and 5 terms of the variable representing node 232 (X_3) are exogenous sources of influence towards node 7 (X_1). On the other hand, the parameter vector for node 232 (X_3) is shown in Figure 12(c). The associated significance indicator grid is shown in Figure 12(d). The causality result suggests that node 7 is also an exogenous source of cause for the anomaly measures of node 232. Therefore, the anomaly



(a) Parameter vector for node 7 (b) Causality indicators



(c) Parameter vector for node 232 (d) Causality indicators

Figure 12: (a) The parameters of all 5 lags for the rVAR model of node 7. (b) The anomaly measures of node 232 Granger cause those of node 7. (c) The parameters for the rVAR model of node 232. (d) The anomaly measures of node 7 Granger cause those of node 232.

measures of node 7 and node 232 are closely correlated.

By checking the original data, we find that user $Jodosma_7$ builds an edge of weight 3 with user $Bnseagreen_{232}$ through times 735256.505 (January 22, 2013 12:07:12. All the other time stamps could be converted to this format), 735256.507 and 735296.924 on the page titled "Dhani Matang Dev". Furthermore, none of the edits made by user 7 and user 232 were reverted. Therefore, both users are considered normal users who have edited the same page. The labels for both users are benign.

6.5.2 Case Study 2

In this case study, we look at user $SegaKing247_9$, who has 2 neighbors, 30 and 666. The adjusted model using stepwise backward elimination multivariate Granger causality analysis method is

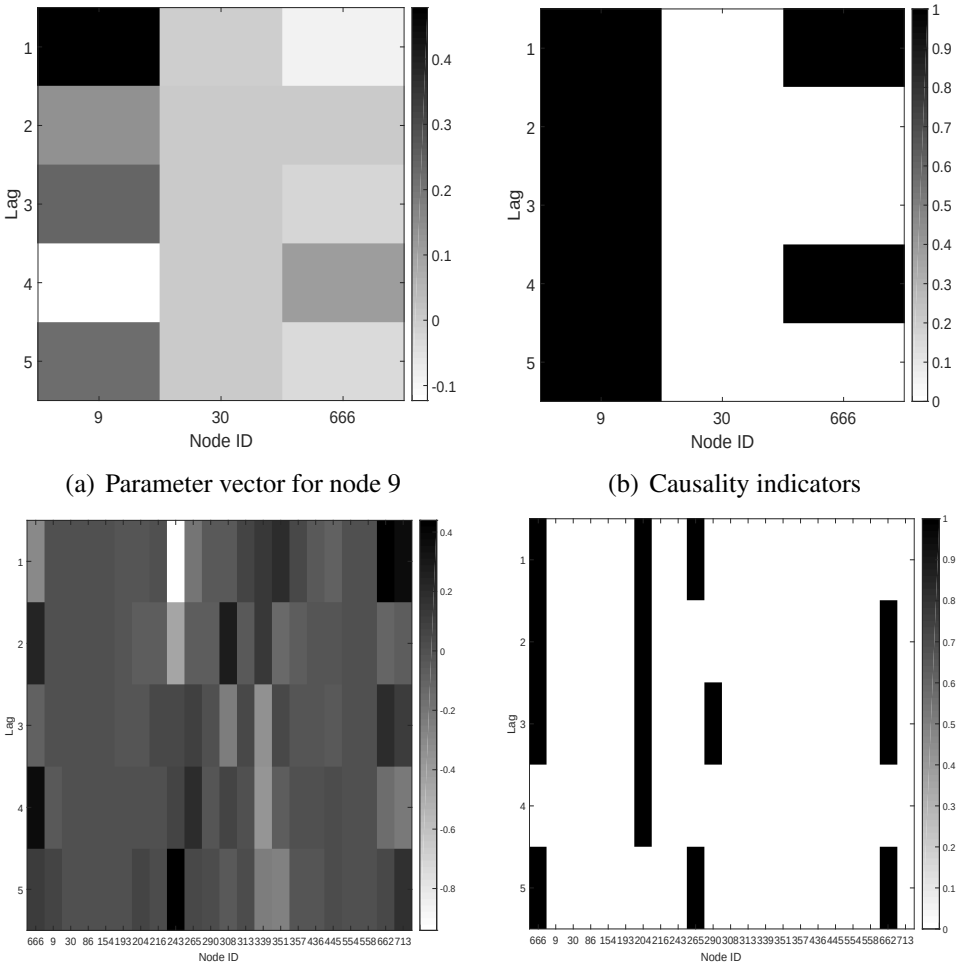
$$\begin{aligned} X_{1,t} = {} & 0.478X_{1,t-1} - 0.085X_{3,t-1} \\ & + 0.134X_{1,t-2} \\ & + 0.235X_{1,t-3} \\ & - 0.122X_{1,t-4} + 0.108X_{3,t-4} \\ & + 0.218X_{1,t-5}, \end{aligned} \tag{41}$$

where $c = 0.00003$ is not significant with t-statistic of -0.27542.

The parameters and Granger causality significance indicators of the fitted rVAR model for users 9 (X_1) and 666 (X_3) are shown in Figure 14. We can see that the causal relationship is one directional from user 666 to user 9. Both users edited the page titled "Sega". By checking the edit history, we find that all of the edits of user 666 were reverted, but only part of edits of user 9 were reverted. Hence, we conclude that the activities of user 66 are abnormal and they influenced the anomaly measures of user 9, while user 9 could be a benign user. The label for user 666 is vandal and the label for user 9 is benign.

6.5.3 Case Study 3

The target node is 466 and its 1-step neighbors are 67, 312, 330, 421, 563, 605 and 683. The node anomaly measure variables are relabeled as X_1 to X_8 respectively. The adjusted model using stepwise backward elimination multivariate Granger causality analysis method



(a) Parameter vector for node 9

(b) Causality indicators

(c) Parameter vector for node 666

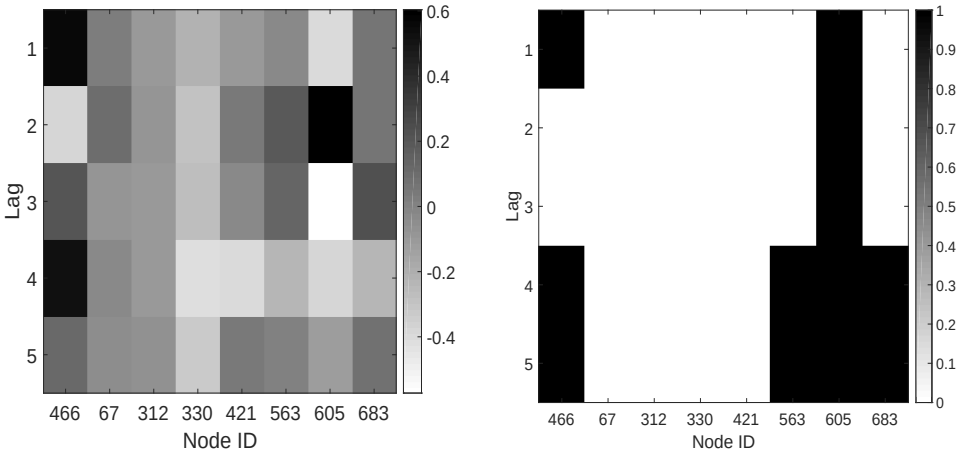
(d) Causality indicators

Figure 13: (a) The parameters of all 5 lags for the rVAR model of node 9. (b) The anomaly measures of node 666 Granger cause those of node 9. (c) The parameters for the rVAR model of node 666. (d) The anomaly measures of node 9 do not Granger cause those of node 666.

is

$$
\begin{aligned}
X_{1,t} = {} & 0.562X_{1,t-1} - 0.396X_{7,t-1} \\
& + 0.603X_{7,t-2} \\
& - 0.572X_{7,t-3} \\
& + 0.517X_{1,t-4} - 0.231X_{6,t-4} - 0.387X_{7,t-4} - 0.229X_{8,t-4} \\
& + 0.119X_{1,t-5} + 0.014X_{6,t-5} - 0.114X_{7,t-5} + 0.083X_{8,t-5},
\end{aligned}
\tag{42}
$$

where $c = 0.033$ is significant with t-statistic value of 2.3034. The parameter vector and the associated from Granger casuality indicators are shown in Figures 14(a) and 14(b).



(a) Parameter vector for node 466

(b) Causality indicators

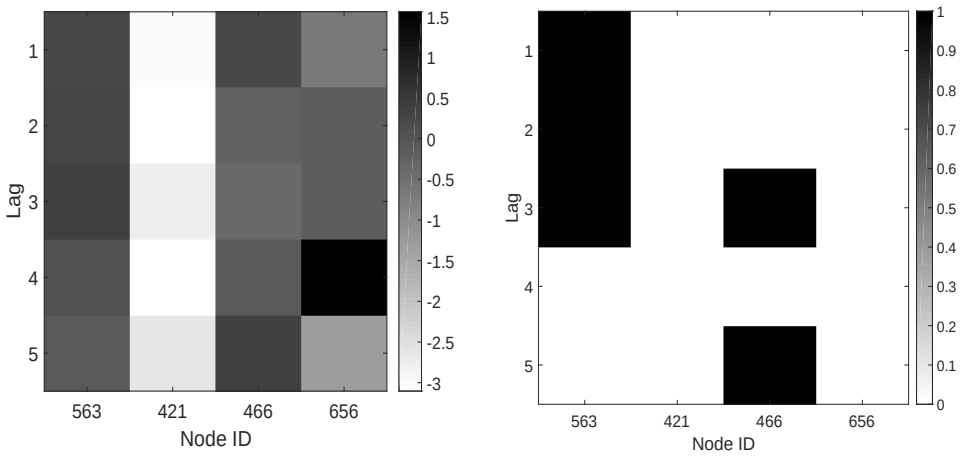
(c) Parameter vector for node 605

(d) Causality indicators

Figure 14: (a) The parameters of all 5 lags for the rVAR model of node 466. (b) The anomaly measures of node 605 Granger cause those of node 466. (c) The parameters for the rVAR model of node 605. (d) The anomaly measures of node 466 Granger cause those of node 605.

In this example, both users $Grobelaar0811_{466}$ (X_1) and $Bobcalderon_{605}$ (X_7) edited the page titled "Sofia Vergara" together. An edge of weight 4 was built between two users through times 735308.6021, 735308.6025, 735355.0022 and 735355.0029. The casuality analysis suggests that the activities of user 563 Granger cause the anomaly measures of

user 466.



(a) Parameter vector for node 563 (b) Causality indicators

Figure 15: (a) The parameters of all 5 lags for the rVAR model of node 563. (b) Node 466 is an exogenous source of influence to the anomaly measures of node 563.

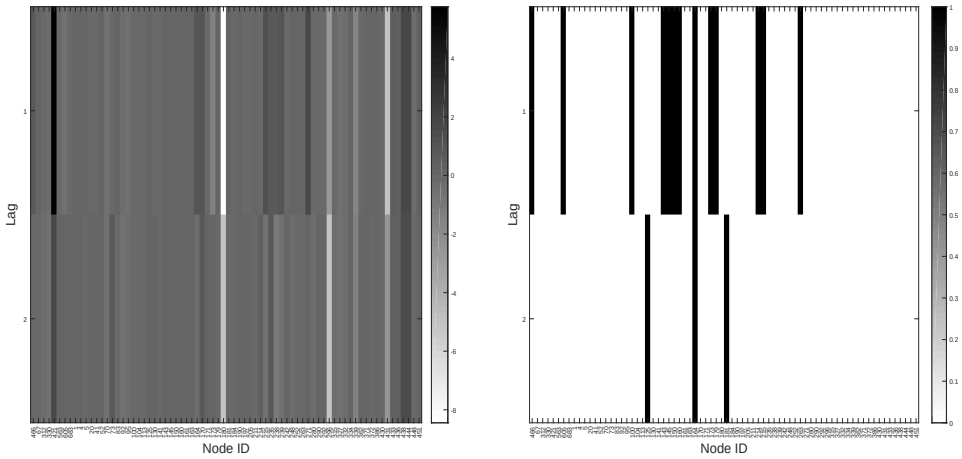
By looking at the model and Granger Causality results for node 605 in Figure 14(c) and Figure 14(d), we notice that the causal relationship between user 466 and user 605 is bidirectional and conclude that they have a relatively close relationship. By further checking the edit history of both users, we find that all 7 edits of user 466 and all 5 edits of user 605 were reverted, which indicate that both users could be vandals with very high probability. Combining all the observations with the causality analysis results, we suspect that user 466 and user 605 may have attacked the page collaboratively. The labels for both users are vandal.

As shown in Figure 15(a) and Figure 15(b), another neighbor, user $Themaxandpeter_{563}$ also has similar causality results with node 466 as node 605 does. Node 466 and node 563 edited two pages titled "Fulham F.C." and "Dynamo (magician)" together. According to the edit history, all of the edits made by user 563 were reverted. According to all the above results, we suspect that all 3 users 466, 563 and 605 were collaborating their attacks on

Wikipedia pages.

6.5.4 Case Study 4

In this case study, we look at the rVAR(2) model of node 466 with its 2-step neighbors. Node 466 has 7 1-step neighbors and 115 2-step neighbors. The meaning for 2-step neighbors in this particular dataset is that those nodes edited some pages together with the 1-step neighbors of the target node. For any relatively well connected graph, the number of 2-step neighbors tends to grow very large. As mentioned before, fitting a rVAR model on a large number of variables requires more computational power and may have rank deficiency problems. After the rank test, only 73 neighbors and the target have time series data that are not linearly dependent. The Johansen cointegration test suggests that all 73 neighbors' time series are cointegrated with the target node's time series. The rVAR model parameters and Granger causality indicators are shown in Figure 16(a) and Figure 16(b).



(a) Parameter vectors for node 466 (b) Causality indicators

Figure 16: (a) The parameters of 2 lags for the rVAR model of node 466 with its 2-step neighbors. (b) The associated Granger causality indicators.

As more variables entering the model, we can see that the causality analysis results become more complicated. Since the model contains more variables, only a few lags could

be incorporated to make the computational time manageable. The results indicate that the majority of exogenous causal influences come from the 2-step neighbors. However, the weights of the influences of any 2-step neighbors should be significantly lowered to reflect their weaker conductivities (in terms of distances) with node 466 than those of the 1-step neighbors. Node 605 is the only 1-step neighbor appearing in the indicator figure, while node 563 and 683 are not captured with the model of lag 2. An important issue is that, if the number of variables entering the model increases, the chance for the model to be overfit will also increase. Therefore, in applications, the number of variables and the size of the lag need to be chosen carefully to prevent overfitting a model and to save computational resources.

6.6 Summary

We have presented a noval approach for modeling the correlations of the node anomaly measures calculated from the spectral features of the dynamic graph generated from a given streaming OSN data by using rVAR model. We have also proposed to use the stepwise backward elimination Granger casuality method to analyze the casual relationships of node activities from the fitted rVAR model. To our knowledge, this is the first work to systematically analyze the graph spectrum based anomaly metric time series data simultaneously using a multivariate statistical modeling tool. This is also the first work to use a strict statistical inference method for identifying the endogenous and/or exogenous sources of casual influence of node interactions in dynamic graphs. As demonstrated in the case studies, by quantifying the randomness of node activities into the node nonrandomness measures and analyzing the resulting time series data, the proposed method could help us identify

different activity patterns such as collaborative attacks, benign users sharing a common interest, and benign users being attacked by vandals. The analysis results from the presented algorithm are visualized to make them easier for users to interpret.

For future works, we could explore the features of the fitted rVAR model and Granger causality results so that they could be used to construct a supervised or unsupervised learning method for node classification. Since the method proposed in this chapter is modularized where different types of anomaly metrics could be plug in to analyze different anomaly behaviors, we will explore different anomaly metrics under different scenarios for a more extensive coverage of anomaly detection. Another important direction is to derive a more sophisticated node nonrandomness measures for the directed signed and weighted graphs using rigorous probabilistic modeling approaches rather than the node weight adjusted version used in this chapter.

CHAPTER 7: CONCLUSION AND FUTURE WORK

7.1 Conclusion

Although the adjacency eigenspace captures rich information about the structure information and node behaviours of a given network, it has not received sufficient attention. The primary reason is that it is challenging to handle complex eigenpairs introduced by the asymmetries of such network data. The asymmetries further cause the non-orthogonormal eigen decompositions of the adjacency matrices, which make studies in such eigenspaces very challenging. In this dissertation work, we proposed a matrix perturbation based theoretical framework and used it to explain several phenomena observed in the adjacency eigenspaces for directed unsigned graphs in Chapter 3 and directed signed graphs in Chapter 4. This framework could serve as a tool to handle the issues caused by the asymmetries of directed networks in general. Based on these theoretical results, we proposed two algorithms Augmented_ADJCluster and General_ADJCluster that could detect clusters in directed unsigned and directed signed networks respectively. The later algorithm could reduce naturally to the former one for directed unsigned graphs.

We also studied the asymmetric information captured by SVD spectral space of the skew symmetric graphs in Chapter 5. Both dominance and submissiveness score measures were proposed to depict the organizational hierarchy of any relationship network. We developed an algorithm that could evaluate the dominance structures of a given network. We also

proposed to use probability distributions of the network dominance structures to compare organizational hierarchies of different relationship networks.

For the anomaly detection and analysis of OSN data, based on the spectral analysis results from all earlier chapters, we proposed in Chapter 6 to use the modified node non-randomness metric derived from the spectral coordinates of the adjacency eigenspaces as the time series data to quantify the changes of the node anomalies. Then, we used rVAR method to model the interactions amongst the individual nodes. The fitted rVAR model could be analyzed using Granger casuality method or CGCI method to identify the endogenous and exogenous sources of anomaly influences. Rather than deriving a threshold at each snapshot of the network as most existing spectral analysis based anomaly detection methods, we studied the anomalies of the entire network simultaneously and systematically as a whole.

The algorithms developed in these chapters were evaluated alongside many state-of-the-art methods. We evaluated them on synthetic data with various structural properties and sizes. In addition, many real dataset such as Sampson's, Slashdot Zoo, Wikisigned, Twitter streaming data, Epinion and World Trade 2014 data were analyzed using our algorithms.

7.2 Future Work

There are still many unexplored aspects of the graph adjacency eigenspace.

For graph theory related topics, there are two important problems. First, further knowledge of signed adjacency matrices outside of the set of Perron Frobenius property is still needed. Second, the geometric meanings of complex eigenpairs in relation to graph structures are still unclear. Those two problems have caused some major challenges for under-

standing the eigenspace properties of the adjacency matrices for directed graphs, especially directed signed graphs. As the number of negative entries increases for a given directed signed graph, the adjacency matrix deviates further from the Perron Frobenius property, but the threshold for which the leading real eigenpair corresponding to a connected cluster vanishes is still unknown. Additional works in these topics using set theory and group theory may be needed to fill in the gaps.

For matrix perturbation theory, there are two topics that could be explored in next step. First, the theoretical framework for describing spectral behaviours of graphs with large perturbations is still absent. Since all current theoretical studies of matrix perturbations are limited to small perturbations, it is unclear how the derived theoretical results will change when the perturbations increase. It is helpful to figure out whether the increase in perturbations will cause the theoretical results to fail or will just increase the errors of the approximations. Second, spectral analysis of DUGs and DSGs presented in Chapters 3-4 could work well for asymmetric weighted graphs in principle. Hence, we can extend the current works to study the spectral properties of directed weighted graphs.

For studying dynamic graphs, there are still many topics that could be explored. First, a complete extension of the node nonrandomness metric into signed weighted graphs could be helpful. Second, deriving a confidence bound from the probabilistic distributions of the anomalous nodes and normal nodes could better aid us to distinguish whether a node is normal or abnormal. Third, logistic regression or similar techniques could be used to provide classifiers using the features extracted from the Granger causality analysis results based on the fitted rVAR model.

For data mining in large streaming datasets, the scalability of the algorithms introduced

in this work could be further explored. For our current studies, we focused on networks within the million-size scale. With the help of a distributed processing system such as Hadoop, we could explore the possibility to parallelize our algorithms so that networks of the billion-size scale could be studied.

Our theoretical results of spectral analysis of directed graphs could be applied to other tasks in addition to community partition, organization hierarchy, and fraud detection. One particular direction of our future work is to extend the spectral analysis based visualization of undirected graphs [48, 49] to directed graphs.

REFERENCES

[1] Wikisigned network dataset – KONECT, Jan. 2016.

[2] B. Abraham and G. E. Box. Bayesian analysis of some outlier problems in time series. *Biometrika*, pages 229–236, 1979.

[3] A. Agovic, A. Banerjee, A. R. Ganguly, and V. Protopopescu. Anomaly detection in transportation corridors using manifold embedding. *Knowledge Discovery from Sensor Data*, pages 81–105, 2008.

[4] M. Agyemang, K. Barker, and R. Alhajj. A comprehensive survey of numeric and symbolic outlier mining techniques. *Intelligent Data Analysis*, 10(6):521–538, 2006.

[5] H. Akaike. Factor analysis and aic. *Psychometrika*, 52(3):317–332, 1987.

[6] M. Al Hasan and M. J. Zaki. A survey of link prediction in social networks. In *Social network data analytics*, pages 243–275. Springer, 2011.

[7] P. Anchuri and M. Magdon-Ismail. Communities and balance in signed networks: A spectral approach. In *Advances in Social Networks Analysis and Mining (ASONAM), 2012 IEEE/ACM International Conference on*, pages 235–242. IEEE, 2012.

[8] M. Aouchiche and P. Hansen. A survey of automated conjectures in spectral graph theory. *Linear Algebra and its Applications*, 432(9):2293–2322, 2010.

[9] A. Arenas, J. Duch, A. Fernandez, and S. Gomez. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9:176, 2007.

[10] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 619–626. ACM, 2001.

[11] Z. A. Bakar, R. Mohemad, A. Ahmad, and M. M. Deris. A comparative study for outlier detection techniques in data mining. In *Cybernetics and Intelligent Systems, 2006 IEEE Conference on*, pages 1–6. IEEE, 2006.

[12] R. H. Bartels and G. W. Stewart. Solution of the matrix equation AX+XB=C. *Comm. ACM*, 15:820–826, 1972.

[13] R. J. Beckman and R. D. Cook. Outlier. s. *Technometrics*, 25(2):119–149, 1983.

[14] C. Berge. *Theorie des graphes et ses applications: 2e ed.* Dunod, 1967.

[15] L. F. Berkman and T. Glass. Social integration, social networks, social support, and health. *Social epidemiology*, 1:137–173, 2000.

[16] A. M. Bianco, M. Garcia Ben, E. Martinez, and V. J. Yohai. Outlier detection in regression models with arima errors using robust estimates. *Journal of Forecasting*, 20(8):565–579, 2001.

[17] A. M. Bruckstein and D. Snaked. Skew symmetry detection via invariant signatures. *Pattern Recognition*, 31(2):181–192, 1998.

[18] P. K. Chan and M. V. Mahoney. Modeling multiple time series for anomaly detection. In *Data Mining, Fifth IEEE International Conference on*, pages 8–pp. IEEE, 2005.

[19] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[20] W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1029–1038. ACM, 2010.

[21] K.-Y. Chiang, J. J. Whang, and I. S. Dhillon. Scalable clustering of signed networks using balance normalized cut. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 615–624. ACM, 2012.

[22] C. Chiu, Y. Ku, T. Lie, and Y. Chen. Internet auction fraud detection using social network analysis and classification tree approaches. *International Journal of Electronic Commerce*, 15(3):123–147, 2011.

[23] L. Chu, Z. Wang, J. Pei, J. Wang, Z. Zhao, and E. Chen. Finding gangs in war from signed networks.

[24] F. Chung. *Spectral graph theory*. Amer Mathematical Society, 1997.

[25] F. Chung. Laplacians and the cheeger inequality for directed graphs. *Annals of Combinatorics*, 9(1):1–19, 2005.

[26] D. Cvetkovic, M. Doob, and H. Sachs. Spectra of graphs-theory and applications, iii revised and enlarged edition. *Johan Ambrosius Bart Verlag, Heidelberg-Leipzig*, 1995.

[27] L. da F. Costa, F. A. Rodrigues, G. Travieso, and P. R. V. Boas. Characterization of complex networks: A survey of measurements. *Advances In Physics*, 56:167, 2007.

[28] I. S. Dhillon, Y. Guan, and B. Kulis. Kernel k-means: spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–556. ACM, 2004.

[29] C. Ding and X. He. Fast algorithm for detecting community structure in networks. *Proc. 21st International Conference on Machine Learning*, pages 29–37, 2004.

[30] B. Efron, T. Hastie, I. Johnstone, R. Tibshirani, et al. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

[31] K. D. Elworthy, Y. Le Jan, and X.-M. Li. On the geometry of diffusion operators and stochastic flows. 1999.

[32] A. J. Fox. Outliers in time series. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 350–363, 1972.

[33] L. C. Freeman. Uncovering organizational hierarchies. *Computational & Mathematical Organization Theory*, 3(1):5–18, 1997.

[34] S. Friedland. On an inverse problem for nonnegative and eventually nonnegative matrices. *Israel Journal of Mathematics*, 29(1):43–60, 1978.

[35] G. F. Frobenius. Über matrizen aus nicht negativen elementen. *Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu Berlin*, pages 456–477, 1912.

[36] D. Gao, M. K. Reiter, and D. Song. Gray-box extraction of execution graphs for anomaly detection. 2004.

[37] A. Gauchy. Recherche sur les polyèdres-premier mémoire. *Journal de lEcole Polytechnique*, 16:66, 1813.

[38] J. F. Geweke. Measures of conditional linear dependence and feedback between time series. *Journal of the American Statistical Association*, 79(388):907–915, 1984.

[39] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, 1996.

[40] J. C. Gower. The analysis of asymmetry and orthogonality. *Recent developments in statistics*, pages 109–123, 1977.

[41] J. C. Gower and B. Zielman. *Some remarks on orthogonality in the analysis of asymmetry*. University of Leiden, Department of Data Theory, 1992.

[42] C. W. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society*, pages 424–438, 1969.

[43] F. Harary and R. Z. Norman. Graph theory as a mathematical model in social science. 1953.

[44] F. Heider. Attitudes and cognitive organization. *The Journal of psychology*, 21(1):107–112, 1946.

[45] V. J. Hodge and J. Austin. A survey of outlier detection methodologies. *Artificial Intelligence Review*, 22(2):85–126, 2004.

[46] S. A. Hofmeyr, S. Forrest, and A. Somayaji. Intrusion detection using sequences of system calls. *Journal of computer security*, 6(3):151–180, 1998.

[47] N.-J. Hsu, H.-L. Hung, and Y.-M. Chang. Subset selection for vector autoregressive processes using lasso. *Computational Statistics & Data Analysis*, 52(7):3645–3657, 2008.

[48] X. Hu, A. Lu, and X. Wu. Spectrum-based network visualization for topology analysis. *IEEE Computer Graphics and Applications*, 33(1):58–68, 2013.

[49] X. Hu, L. Wu, A. Lu, and X. Wu. Block-organized topology visualization for visual exploration of signed networks. In *IEEE International Conference on Data Mining Workshop, ICDMW 2015, Atlantic City, NJ, USA, November 14-17, 2015*, pages 652–659, 2015.

[50] T. Idé and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 440–449. ACM, 2004.

[51] T. Inohara. Characterization of clusterability of signed graph in terms of newcombs balance of sentiments. *Applied Mathematics and Computation*, 133:93–104, 2002.

[52] S. Johansen. Estimation and hypothesis testing of cointegration vectors in gaussian vector autoregressive models. *Econometrica: Journal of the Econometric Society*, pages 1551–1580, 1991.

[53] Y. Kim, S.-W. Son, and H. Jeong. Finding communities in directed networks. *Physical Review E*, 81(1):016103, 2010.

[54] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.

[55] R. Koenker. Quantile regresssion. *Encyclopedia of Environmetrics*, 2006.

[56] S. Kumar, F. Spezzano, and V. Subrahmanian. Vews: A wikipedia vandal early warning system. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 607–616. ACM, 2015.

[57] J. Kunegis. Handbook of network analysis [konect–the koblenz network collection]. *arXiv preprint arXiv:1402.5500*, 2014.

[58] J. Kunegis and A. Lommatzsch. Learning spectral graph transformations for link prediction. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 561–568. ACM, 2009.

[59] J. Kunegis, A. Lommatzsch, and C. Bauckhage. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web*, pages 741–750. ACM, 2009.

[60] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *SDM*, volume 10, pages 559–559. SIAM, 2010.

[61] J. R. Lee, S. Oveis Gharan, and L. Trevisan. Multi-way spectral partitioning and higher-order cheeger inequalities. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pages 1117–1130. ACM, 2012.

[62] E. A. Leicht and M. E. Newman. Community structure in directed networks. *Physical review letters*, 100(11):118703, 2008.

[63] J. Leskovec, D. Huttenlocher, and J. Kleinberg. Signed networks in social media. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1361–1370. ACM, 2010.

[64] S.-A.-J. L'Huillier. Mèmoire sur la polyèdromètrie. *Annales de Mathèmatiques*, 3:169–189, 1861.

[65] Y. Li, X. Wu, and A. Lu. Analysis of spectral space properties of directed graphs using matrix perturbation theory with application in graph partition. In *Data Mining (ICDM), 2015 IEEE International Conference on*, pages 847–852. IEEE, 2015.

[66] Y. Li, X. Wu, and A. Lu. On spectral analysis of directed signed graphs. In *IEEE International Conference on Data Science and Advanced Analytics, DSAA, Tokyo, Japan, Oct 19-21,*, 2017.

[67] Y. Li, X. Wu, and S. Yang. Social network dominance based on analysis of asymmetry. In *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pages 146–151. IEEE, 2016.

[68] D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

[69] E. Liberty. Simple and deterministic matrix sketching. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 581–588. ACM, 2013.

[70] H. Lütkepohl. Vector autoregressive models. In *International Encyclopedia of Statistical Science*, pages 1645–1647. Springer, 2011.

[71] F. D. Malliaros and M. Vazirgiannis. Clustering and community detection in directed networks:a survey. *Physics Reports*, 553:95–142, 2013.

[72] M. Meila and W. Pentney. Clustering by weighted cuts in directed graphs. *SIAM DM*, PR127:10, 2007.

[73] C. D. Meyer. *Matrix Analysis and Applied Linear Algebra*. SIAM, 2001.

[74] P. Mika. Social networks and the semantic web. In *Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, pages 285–291. IEEE Computer Society, 2004.

[75] P. Mika. Flink: Semantic web technology for the extraction and analysis of social networks. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2):211–223, 2005.

[76] A. Mirzal and M. Furukawa. Eigenvectors for clustering: Unipartite, bipartite, and directed graph cases. *Electronics and Information Engineering*, 1:303–309, 2010.

[77] M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167, 2003.

[78] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[79] M. E. Newman and E. A. Leicht. Community structure in directed networks. *PRL*, 100:118703, 2007.

[80] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74:036104, 2006.

[81] M. E. J. Newman. Modularity and community structure in networks. *Proc Natl Acad Sci*, 103(23):8577–8582, 2006.

[82] A. Y. Ng, M. I. Jordan, Y. Weiss, et al. On spectral clustering: Analysis and an algorithm. *Advances in neural information processing systems*, 2:849–856, 2002.

[83] V. Nicosia, G. Mangioni, V. Carchiolo, and M. Malgeri. Extending the definition of modularity to directed graphs with overlapping communities. *Journal of Statistical Mechanics*, 03:3024, 2009.

[84] H. Ning, W. Xu, Y. Chi, Y. Gong, and T. Huang. Incremental spectral clustering with application to monitoring of evolving blog communities. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 261–272. SIAM, 2007.

[85] C. C. Noble and D. J. Cook. Graph-based anomaly detection. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 631–636. ACM, 2003.

[86] D. Noutsos. On perron–frobenius property of matrices having some negative entries. *Linear Algebra and its Applications*, 412:132–153, 2006.

[87] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.

[88] S. Pandit, D. H. Chau, S. Wang, and C. Faloutsos. Netprobe: a fast and scalable system for fraud detection in online auction networks. In *Proceedings of the 16th international conference on World Wide Web*, pages 201–210. ACM, 2007.

[89] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: A probabilistic analysis. In *Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of database systems*, pages 159–168. ACM, 1998.

[90] A. Patcha and J.-M. Park. An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, 51(12):3448–3470, 2007.

[91] B. A. Prakash, A. Sridharan, M. Seshadri, S. Machiraju, and C. Faloutsos. Eigenspokes: Surprising patterns and scalable community chipping in large graphs. In *PAKDD*, 2010.

[92] R. A. Reyment. Asymmetry analysis of geologic homologues on both sides of the strait of gibraltar. *Journal of the International Association for Mathematical Geology*, 13(6):523–533, 1981.

[93] F. S. Roberts. *Graph theory and its applications to problems of society*. SIAM, 1978.

[94] S. F. Sampson. *Crisis in a Cloister*. PhD thesis, PhD Thesis. Cornell University, Ithaca, 1969.

[95] V. Satuluri and S. Parthasarathy. Symmetrizations for clustering directed graphs. In *EDBT*, pages 343–354. ACM, 2011.

[96] J. A. Saunders and D. C. Knill. Perception of 3d surface orientation from skew symmetry. *Vision research*, 41(24):3163–3183, 2001.

[97] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1046–1054. ACM, 2011.

[98] S. E. Schaeffer. Graph clustering. *Computer Science Review*, 1(1):27 – 64, 2007.

[99] J. Scott. Social network analysis: developments, advances, and prospects. *Social network analysis and mining*, 1(1):21–26, 2011.

[100] J. Scott. *Social network analysis*. Sage, 2012.

[101] A. Seary and W. Richards. Spectral methods for analyzing and visualizing networks: an introduction. *National Research Council, Dynamic Social Network Modelling and Analysis: Workshop Summary and Papers*, pages 209–228, 2003.

[102] S. Shekhar, C.-T. Lu, and P. Zhang. Detecting graph-based spatial outliers: algorithms and applications (a summary of results). In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 371–376. ACM, 2001.

[103] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000.

[104] C. Shih. *The Facebook era: Tapping online social networks to build better products, reach new audiences, and sell more stuff*. Prentice Hall, 2009.

[105] M.-L. Shyu, S.-C. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. Technical report, DTIC Document, 2003.

[106] X. Song, M. Wu, C. Jermaine, and S. Ranka. Conditional anomaly detection. *Knowledge and Data Engineering, IEEE Transactions on*, 19(5):631–645, 2007.

[107] G. W. Stewart. Error bounds for approximate invariant subspaces of closed linear operators. *SIAM Journal on Numerical Analysis*, 8:796–808, 1971.

[108] G. W. Stewart. On the early history of the singular value decomposition. *SIAM review*, 35(4):551–566, 1993.

[109] G. W. Stewart and J. Sun. *Matrix Perturbation Theory*. Academic Press, 1990.

[110] S. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.

[111] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *ICDM*, pages 418–425, 2005.

[112] J. Sun, Y. Xie, H. Zhang, and C. Faloutsos. Less is more: Compact matrix representation of large sparse graphs. In *Proceedings of 7th SIAM International Conference on Data Mining*, 2007.

[113] M. Swan. Emerging patient-driven health care models: an examination of health social networks, consumer personalized medicine and quantified self-tracking. *International journal of environmental research and public health*, 6(2):492–525, 2009.

[114] P. Symeonidis, N. Iakovidou, N. Mantas, and Y. Manolopoulos. From biological to social networks: Link prediction based on multi-way spectral clustering. *Data & Knowledge Engineering*, 87:226–242, 2013.

[115] J. Tang, Y. Chang, C. Aggarwal, and H. Liu. A survey of signed network mining in social media. *arXiv:1511.07569 [physics]*, 2015.

[116] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[117] V. Traag and J. Bruggeman. Community detection in networks with positive and negative links. *Physics Review E*, 80, 2009.

[118] W. F. Trench. Characterization and properties of matrices with generalized symmetry or skew symmetry. *Linear algebra and its applications*, 377:207–218, 2004.

[119] M. Trusov, R. E. Bucklin, and K. Pauwels. Effects of word-of-mouth versus traditional marketing: findings from an internet social networking site. *Journal of marketing*, 73(5):90–102, 2009.

[120] R. S. Tsay, D. Peña, P. Galeano, et al. Outlier detection in multivariate time series via projection pursuit. Technical report, Universidad Carlos III de Madrid. Departamento de Estadística, 2004.

[121] R. S. Tsay, D. Peña, and A. E. Pankratz. Outliers in multivariate time series. *Biometrika*, 87(4):789–804, 2000.

[122] S. P. V. Satuluri. Symmetrizations for clustering directed graphs. *EDBT*, pages 343–354, 2011.

[123] R. S. Varga. *Matrix Iterative Analysis*. Springer, 2009.

[124] F. B. Viégas and J. Donath. Social network visualization: Can we go beyond the graph. In *Workshop on social networks, CSCW*, volume 4, pages 6–10, 2004.

[125] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[126] C. Warrender, S. Forrest, and B. Pearlmutter. Detecting intrusions using system calls: Alternative data models. In *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*, pages 133–145. IEEE, 1999.

[127] N. Wiener. The theory of prediction. *Modern mathematics for engineers*, 1:125–139, 1956.

[128] L. Wu, X. Wu, A. Lu, and Y. Li. On spectral analysis of signed and dispute graphs. In *2014 IEEE International Conference on Data Mining, ICDM 2014, Shenzhen, China, December 14-17, 2014*, pages 1049–1054, 2014.

[129] L. Wu, X. Wu, A. Lu, and Y. Li. On spectral analysis of signed and dispute graphs: Application to community structure. *IEEE Transactions on Knowledge and Data Engineering*, 29(7):1480–1493, July 2017.

[130] L. Wu, X. Wu, A. Lu, and Z. Zhou. A spectral approach to detecting subtle anomalies in graphs. *J. Intell. Inf. Syst.*, 41(2):313–337, 2013.

[131] L. Wu, X. Ying, X. Wu, A. Lu, and Z.-H. Zhou. Spectral analysis of k-balanced signed graphs. In *PAKDD (2)*, pages 1–12, 2011.

[132] L. Wu, X. Ying, X. Wu, and Z.-H. Zhou. Line orthogonality in adjacency eigenspace with application to community partition. In *IJCAI*, pages 2349–2354, 2011.

[133] X. Ying, L. Wu, and X. Wu. A spectrum-based framework for quantifying randomness of social networks. *IEEE Trans. Knowl. Data Eng.*, 23(12):1842–1856, 2011.

[134] X. Ying and X. Wu. On randomness measures for social networks. In *SDM*, 2009.

[135] X. Ying, X. Wu, and D. Barbará. Spectrum based fraud detection in social networks. In *2011 IEEE 27th International Conference on Data Engineering*, pages 912–923. IEEE, 2011.

[136] R. K. Yip. A hough transform technique for the detection of reflectional symmetry and skew-symmetry. *Pattern Recognition Letters*, 21(2):117–130, 2000.

[137] B. G. Zaslavsky and B.-S. Tam. On the jordan form of an irreducible matrix with eventually nonnegative powers. *Linear Algebra and its Applications*, 302:303–330, 1999.

[138] T. Zaslavsky. Matrices in the theory of signed simple graphs. *Proc. Int. Conf. Discrete Math., ICDM*, 2008.

[139] K. Zhang, I. W. Tsang, and J. T. Kwok. Improved nyström low-rank approximation and error analysis. In *Proceedings of the 25th international conference on Machine learning*, pages 1232–1239. ACM, 2008.

[140] Q. Zheng and D. Skillicorn. Spectral embedding of signed networks. In *SIAM International Conference on Data Mining*, pages 55–63. SIAM, 2015.

[141] D. Zhou and C. J. C. Burges. Spectral clustering and transductive learning with multiple views. *ICML*, pages 1159 – 1166, 2007.

[142] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *ICML*, pages 1036–1043. ACM, 2005.

[143] D. Zhou, B. Schölkopf, and T. Hofmann. Semi-supervised learning on directed graphs. In *NIPS*, 2005.