SPECTRUM-BASED AND COLLABORATIVE NETWORK TOPOLOGY ANALYSIS AND VISUALIZATION

by

Xianlin Hu

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

2013

Approved by:

____

Dr. Aidong Lu

____

Dr. Xintao Wu

____

Dr. Jing Yang

____

Dr. Zachary Wartell

____

Dr. Yanqing Sun

ABSTRACT

XIANLIN HU. Spectrum-based and collaborative network topology analysis and visualization . (Under the direction of DR. AIDONG LU)

Networks are of significant importance in many application domains, such as World Wide Web and social networks, which often embed rich topological information. Since network topology captures the organization of network nodes and links, studying network topology is very important to network analysis. In this dissertation, we study networks by analyzing their topology structure to explore community structure, the relationship among network members and links as well as their importance to the belonged communities. We provide new network visualization methods by studying network topology through two aspects: spectrum-based and collaborative visualization techniques.

For the spectrum-based network visualization, we use eigenvalues and eigenvectors to express network topological features instead of using network datasets directly. We provide a visual analytics approach to analyze unsigned networks based on recent achievements on spectrum-based analysis techniques which utilize the features of node distribution and coordinates in the high dimensional spectral space. To assist the interactive exploration of network topologies, we have designed network visualization and interactive analysis methods allowing users to explore the global topology structure.

Further, to address the question of real-life applications involving of both positive and negative relationships, we present a spectral analysis framework to study both signed and unsigned networks. Our framework concentrates on two problems of network analysis - what are the important spectral patterns and how to use them to study signed networks. Based on the framework, we present visual analysis methods, which guide the selection of k-dimensional spectral space and interactive exploration of network topology.

With the increasing complexity and volume of dynamic networks, it is important to adopt strategies of joint decision-making through developing collaborative visualization approaches. Thus, we design and develop a collaborative detection mechanism with matrix visualization for complex intrusion detection applications. We establish a set of collaboration guidelines for team coordination with distributed visualization tools. We apply them to generate a prototype system with interactions that facilitates collaborative visual analysis.

In order to evaluate the collaborative detection mechanism, a formal user study is presented. The user study monitored participants to collaborate under co-located and distributed collaboration environments to tackle the problems of intrusion detection. We have observed participants' behaviors and collected their performances from the aspects of coordination and communication. Based on the results, we conclude several coordination strategies and summarize the values of communication for collaborative visualization.

Our visualization methods have been demonstrated to be efficient topology exploration with both synthetic and real-life datasets in spectrum-based and collaborative exploration. We believe that our methods can provide useful information for future design and development of network topology visualization system.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

Social networks are of significant importance in many application domains, such as marketing, psychology, epidemiology and homeland security. These networks often embed rich topological information, such as community structures, which has been a popular research problem during the past years. The topology structure indicates the arrangements of network nodes and links [60]. There are two basic categories of network topologies: physical and logical topology [121]. Physical topology indicates location of nodes and links for networks as layouts. Logical topology, in contrast, represents data flows from one node to another node in networks. Visualizing and navigating these network topologies are crucial to understand various networks. In this dissertation, we study networks by analyzing their physical topology to explore community structure, the relationship among network members and links as well as their importance to the belonged communities.

However, when the network complexity is increasing, it is extremely challenging to visualize and explore network topology directly and efficiently. Researchers start to use explicit functions of spectrum and eigenvectors to express network topological features. It has been shown that eigenvalues of a network are intimately connected to many important topological features [115]. For example, the eigenvalues of an adjacency matrix encode information about the cycles of a network as well as its diameter. The maximum degree, the chromatic number, the clique number, and the extent of branching in a connected graph are all related to the largest eigenvalue (λ_1). In Wang et al. [130], the authors have studied how a virus propagates in a real work and proved that the epidemic threshold for a network is closely related to λ_1. Estrada and Rodrguez-Velzquez [43] have shown that the subgraph centrality (SC), which characterizes the participation of each node in all subgraphs in a network, can be calculated mathematically from the spectra of the adjacency matrix of the network. In the first part of this dissertation, we study useful spectrum features and explore how to incorporate spectrum features to develop topology-aware network visualization.

When the network datasets are large-scale and time-varying, the difficulty for the visual exploration of topology structure is increased as well. The amount of information from these networks cannot be analyzed efficiently by a single analyst. Therefore, collaborative problem-solving has started to attract interests of visualization researchers. Actually, collaborative analysis can benefit many large scale applications where a small group of users discuss and negotiate their interpretations of the data with which they are working [58, 65, 74]. On the other hand, visualization can also benefit from collaborative analysis as well. For example, Bresciani and Eppler [17] analyzed the impact of visualization on knowledge sharing in situated work groups and showed that interactive visualization could bring positive and productive changes for group work. In the second part of this dissertation, we analyze collaborative problem-solving features in network visualization and propose a collaborative visualization framework to improve the efficiency of network topology exploration for large-scale and time-varying datasets.

Many researches proved that well-designed visualization can provide useful visual information of networks, which can often be easily perceived by human beings, thus promoting effective and efficient visual network analysis approaches [28], such as our spectrum-based visual analytics approach. The goal of our approaches is to study and visualize network topology including community structure, node-link relationship and importance. We transfer the network datasets into a high-dimensional spectral space, and then apply the high-dimensional spectral features to study network topology. Finally, the network topology in high-dimensional space is visualized in a two-dimensional plane. Additionally, our collaborative framework by studying the collaboration of multiple users working in the application of network visualization proposes a way for groups of people to interact with visualization better. We believe our approaches can propose informative visual analytics methods, intuitive visualization and efficient visual exploration, which are very important to network topology analysis.

1.1 Overview

We presents effective approaches of network visualization and analysis mainly from two aspects: spectrum-based and collaborative network exploration. Chapters 3 and 4 describe our spectrum-based network topology exploration approach. Chapters 5, 6 and 7 present our design and development of a collaborative visualization mechanism for multiple users in dynamic network environments to improve the accuracy and efficiency of collaborative network analysis.

1.1.1 Spectrum-based network topology exploration

We start from studying the theoretical spectral features, then, we visualize the network topology based on these spectral features.

Chapter 3 presents our spectrum-based network visualization approach for analyzing unsigned network topologies, which includes both the global topology structure and the importance of individual nodes and edges to each network community. This approach provides both an automatic network layout algorithm and several interactive analysis tools for analyzing network topologies.

Chapter 4 presents a spectral analysis framework to study both signed and unsigned networks because many real-life applications involve of complex networks containing both positive and negative relationships. We first have a brutal experiment to study the changes of spectral features for signed network with negative edges. Then, based on the results of the experiment, we propose a framework to concentrate on two problems of network analysis - what are the important spectral patterns and how to use them to study signed networks. We have explored community structures for special signed networks. Based on the framework, we present visual analysis methods, which guide interactive exploration of network topology for users.

1.1.2 Collaborative network exploration

Chapter 5 presents our strategies of network topology pattern visualization for detecting malicious attacks in dynamic network environments. Our strategies concentrate on assisting users to analyze statistical network topology patterns that could expose significant attack features. Specifically, we investigate *Sybil attacks* that have severe impacts on the fundamental operations of wireless networks. We have analyzed the features of network topologies under various Sybil attacks and consequently designed several matrix reordering algorithms to generate statistical patterns.

Chapter 6 introduces our collaborative mechanism with above network topology pattern visualization based on a network security analysis application. We first analyze the challenges and requirements for designing such a collaborative mechanism combining the knowledge from human behavior, social aspects and teamwork theory. Then, we describe a web-based prototype system, which is built based on our design principles and heuristics. Our system supports multi-user input, shared and individual views on findings, and flexible workspace organization to facilitate group analysis.

Chapter 7 describes our evaluation on how users collaborate under different collaboration environments to tackle the problems in network security applications. We observe participants behaviors and collect their performances from the aspects of coordination and communication. Based on the results, we conclude several coordination strategies and summarize the values of communication for collaborative detection.

1.2 Contribution

The dissertation presents several approaches for network topology analysis in different perspectives. The main contributions of the dissertation are the following:

By taking advantage of spectrum geometry features of unsigned and signed networks in high-dimensional spaces, important network topology structure can be explored effectively and efficiently by our non-iterative spectrum-based approaches. We also provide visual analysis functions to analyze both the global network topology and the roles of individual nodes and edges to their communities, which are previously hard-to-obtain. The efficiency of our spectrum-based approach is rooted from collecting network topology features in high-dimensional spectral spaces and visualize the features by wrapping the high-dimensional spaces to two-dimensional spaces. This new data transformation based on the spectrum-based data analysis is significantly different from the other force or energy driven methods. Therefore, the main contribution of our spectrum-based approaches is to combine theoretical spectral analysis with interactive visualization methods to study topology structure of both unsigned and signed networks, which have been demonstrated to be effective with real-life and synthetic network datasets.

By applying social aspects and teamwork theory, our efficient collaborative mechanism based on dynamic network security analysis is designed, which to our knowledge has not been explored in significant depth. The evaluation study explores and summarizes collaboration strategies for network security applications. Our results can provide guidelines for future design and development of collaborative network security visualization systems. The coordination and communication strategies can also be used for general collaborative visualization design.

CHAPTER 2: RELATED WORK

2.1 Network Analysis

Network analysis can get insight into the structure information such as communities, nodes behavior, and remote collaboration between nodes. Usually, network analysts use a combination of metrics and visualization to explore network features [127]. In this chapter, we get an overview of network visualization and introduce spectrum-based network analysis, which are related with our approaches.

2.1.1 Network Visualization

The techniques of network visualization have been applied to various fields, including personal social networks [53], citation networks [70], network security [37, 45, 91, 55, 76, 112, 71, 92], biological networks [2, 63], and the World Wide Web [32]. The research topics of network visualization are ranging from aesthetically visualization, topology exploration and navigation, large-scale network analysis to network visualization application. In this chapter, we review literature in visualization and graph drawing fields and get an overview of network visualization.

In the topic of aesthetically network visualization, the generation of network layout is very important because it provides the foundation for network visual analysis and exploration techniques. Popular network layouts include node-links, space division, space nested visualization, and matrix visualization [28]. The algorithms of node-link visualization can be categorized according to the layouts they generate [28], such as tree layouts [102, 18, 110], spring layouts [39, 47, 50, 40], and tree+link layouts [35, 77, 68]. Other categories of node-link network are based on the models they applied. For example, the force-directed approaches [47, 8, 11, 22, 44, 84] often simulate a network as a physical system where edges are springs and nodes are particles, which are the most popular approaches in network layouts [117]. Space division visualization uses the space efficiently to represents parents, children and siblings relationships [4, 118]. Space nested visualization presents hierarchical structure of networks in a space-filling approach, such as Treemaps [78]. These visualization have also been extended to three-dimensional spaces [69, 88, 111, 113, 123]. Matrix visualization presents the networks based on their connections. It is suitable to large or dense networks with its quick layout and great readability [52], which has been widely used to study network topology. Recently, these traditional network layouts have been improved to visualize complex datasets. For example, the node-based techniques [5, 97] usually improve the layouts by reorganizing the position of the nodes while edge-based techniques [29, 101, 107, 108, 106] focus on reducing the visual clutter by dealing with the edges. All of these work focus on generating aesthetical and intuitive network layouts.

Network topology exploration is another popular research topic. For example, Noack [103] proposed a LinLog method to explore the community structures of networks through minimizing an energy model iteratively. Gronemann and Jünger [59] provided a visualization to show how clustered graphs could be drawn as topographic maps. Didimo et. al [34] studied the problems of designing graph drawing algorithms within two heuristics: topology-driven and force-directed. Greffard et. al [56] made a comparison for two-dimensional and three-dimension perspective on community detection in networks. In this dissertation, the focus of our network analysis is also to visualize network topology. But the entry point of our analysis is network feature transformation from high-dimensional spectral space to two-dimensional plane, which is different from previous approaches.

Additionally, interaction techniques have been also developed with visualization to analyze network structure. Because interactions, such as selection, zooming, and editing, are simple and powerful to understand a network and to explore its hidden multiple interpretations [67]. For example, Chan et al. [21] provided a set of interaction methods for analyzing network hierarchies. Arvo [6] gave a review about different interactive graph drawing algorithms. There were many interactive toolkits or systems of network visualization, such as WiGis [57], Gravisto [9], Gluskap [38], HGV [109] and RINGS [124].

Moreover, researchers have proposed algorithms to design network visualization for large-scale and dynamic datasets. Some researchers focused on how to accelerate the speed for large-scale networks exploration. For example, Brandes and Pich [16] presented a novel sampling-based approximation technique for classical MDS that yields an extremely fast layout algorithm suitable even for very large graphs. Some other researches focused on providing methods to analyze hierarchical structure or clusters of large networks. For example, Hong and Murtaph [69] proposed a new method for visualization of large and complex networks in three dimensions. They visualized the core tree structure of large and complex networks hierarchy (sub trees). Additionally, some new network visualizations have been developed by improving the classic force-directed approaches. For example, a system named by Graphael was developed by combining several traditional force-directed layouts to study large networks [46]. A new layout paradigm for drawing large network with a focus on decompositional properties was proposed in [54]. The technique worked in three phases: abstract layout, drawing for individual nodes, and final layout by force-directed methods. In [62], a new force-directed graph drawing algorithm combining with a multilevel scheme and a force model was presented.

For dynamic networks, their structures are evolving over time, which contain more uncertainty than traditional networks. Sallaberry et. al [114] proposed a new approach to cluster, visualize and navigate dynamic networks. They provided dual visual representation views: an overview to show the changes of clusters over time with selection interaction and a node link diagram to visualize the network clusters in a selected time step. Usually, dynamic network analysis has been involved into application. For example, Boitmanis et. al [13] designed a visualization for large dynamic autonomous-level internet topology to analyze internet evolution. Suntinger et. al [120] provided a event-tunnel visualization to find the stream of events in business by backtracing business incidents and exploring event patterns root causes. They also allowed users to search relevant events within a data repository.

After review hundreds of literature on network visualization, we find that intuitive visualization can help people get a meaningful overview of networks as well as the relationship of nodes and edges in detail, which is very helpful for network analysis.

2.1.2 Signed Network Analysis

There are many researches of network visualization and analysis focusing on networks with only positive links. However, in reality, many networks are signed graphs with negative links. For these signed networks, researches of network visualization and visual analytics have focused on visualizing conflicts or controversy relationships in social or political networks. For example, Brandes et al. [14] presented a visual summary method for bilateral conflict structures embodied in event data. This method projected nodes into a conflict space to provide a graph overview and highlight main opponents in a series of events. Brandes and Lerner [15] later proposed a visual analysis technique to reveal authors that were the most involved in controversy and identify some recurrent patterns of confrontation.Suh et al [119] described a model for identifying patterns of conflicts in Wikipedia articles based on users' editing history and relationships between user edits, such as reverts - revisions that void previous edits. The overall conflict patterns between groups of users were also visualized with Revert Graph. Recently, Kermarrec and Moin [80] presented Signed LinLog model based on Linlog model whose clustering properties for unsigned graphs was already discovered. Their method was based on a dual energy model for graphs containing uniquely negative edges, which extended previous energy models for unsigned graphs.

Signed networks have also been studies in a number of other research fields, such as data mining, human-computer interaction, WWW, agent and multi-agent systems, and social network analysis. Sharma et al. [135] proposed a clustering re-clustering algorithm to mine signed social networks where the negative inter-community links and the positive intra-community links are dense. The algorithm first formed clusters using only positive links and then modified the clusters on the basis of a robust criteria termed as participation level. Kunegis et al. [86] adopted social network analysis techniques to the problem of negative edges and studied the Slashdot technology news site. Bogdanov et al. [12] presented a framework for discovery of collaborative community structure in Wiki-based knowledge repositories. Their approach included modeling of pairwise variable-strength contributor interactions and synthesis of a global network incorporating all interactions.

2.1.3 Spectrum-based Network Analysis

In applied mathematics and scientific computing, the spectral methods study graphs using methods of linear algebra, which have been demonstrated useful as one part of graph theory [23]. Especially for network analysis, spectral-related researches have been done to analyze and describe the network properties and the node relationships. For example, Seary and Richards [115] used the spectral methods to discover the cohesive and localized features of networks. Newman [99] used eigenspectrum of a matrix to detect community structure in networks. Ying and Wu[138] studied several important topological features of the network data by preserving spectral properties during randomization. The data features explored in spectral spaces have also be integrated to improve network layouts. For example, Koren etc. [83] provided one algorithm named ACE to draw graphs recursively by eigenvectors of the Laplacian matrix.

The theories for signed graphs, such as balance theory and clusterizable graphs, have been used in various studies about social networks [90]. The first of these theories is structural balance theory, which originated in social psychology and formulated by Heider in the 1940s [66]. Cartwright and Harary introduced this concept to graph theory in the fifties [19] and used it to characterize forbidden patterns for social networks. Informally, a balanced signed graph is a signed graph that respects the following social rules: my friend's friend is my friend, my friend's enemy is my enemy, my enemy's friend is my enemy, and my enemy's enemy is my friend. The authors in [132, 133]showed that communities in a balanced signed graph are distinguishable in the spectral space of its signed adjacency matrix even if connections between communities are dense. Davis in [31] gave a second characterization for social networks by introducing the notion of clusterizable graph. A signed graph is clusterizable if it shows a clustering such that each positive edge connects two vertices of the same subset and each negative edge connects vertices from different subsets. Empirical studies on real data with large databases of social networks have been performed [90, 122] and proved that they fail to reflect some current practices in real social networks [81].

Other approaches have also been explored for graph drawing and mining problems. For example, Leskovee et al. [81] presented an approach to decide whether a given signed graph has a drawing in a given l-dimensional Euclidean space, which satisfies the requirement that everybody sits closer to their friends than their enemies. Kunegis et al. [87] presented spectral graph mining algorithms for signed graphs. Their work demonstrated that several characteristic matrices of graphs could be extended to signed graphs, such as a signed variant of the graph Laplacian. They concentrated on spectral clustering and graphs with positive connections inside communities and negative connections between communities. Leskovee et al. [90] examined the interplay between positive and negative links in social media. They analyzed two theories of signed networks, balance and status, and applied these two theories of connections to make predictions of the other connections.

2.2 Collaborative Visualization

Collaborative visualization systems have been developed for large scientific projects [93] as well as information visualization [17]. Here, we briefly describe the previous work on distributed and co-located collaborative visualization. Later, we introduce the mechanics and social aspects in collaboration.

2.2.1 Distributed Collaborative Visualization

Distributed environment is a popular setting for collaborative visualization. It allows users from different locations to access data remotely and share data analysis results. For example, Ma and Wang [93] provided an example of web-based system, which displayed the results of each simulation run in terms of visualization, animation, and notes made by those who examined the simulation results. Susan et al. [41] built a collaborative framework among multiple analysts, which allowed multiple analysts to share information, especially the reasoning behind the information, logically and graphically. Convertino et.al. [26] investigated the knowledge sharing issue for multiple views with distributed and synchronous visualization.

2.2.2 Co-located Collaborative Visualization

Different from distributed systems, co-located collaborative visualization provides a platform for users to communicate effectively at the same location, such as around a touch table or a multi-display environment. For example, Isenberg and Carpendale [74] presented a system to facilitate hierarchical tree comparison tasks for a small groups using a shared interactive tabletop display. They provided an analysis of challenges and requirements for the design of co-located collaborative information visualization systems. Isenberg et al. [75] made an evaluation for co-located visual analysis around a tabletop display. They concluded eight collaboration styles for co-located visual analytics problem solving. Waldner et al. [128] discussed design considerations for employing multiple-view visualizations in collaborative multi-display environments. Park et al. [105] explored collaboration issues for a CAVE-based virtual reality environment.

2.2.3 The Mechanics and Social Aspects of Collaboration

Collaborative teamwork includes two important components: the mechanics and social aspects of collaboration. The mechanics of collaboration include common actions which team members must take to complete a shared task in the collaboration process. For example, Gutwin and Greenberg [61] identified several major actions including communication, coordination, planning, monitoring, assistance, and protection.

Ma and Wang [93] have pointed out that knowledge sharing and the social aspects of collaboration should be considered; particularly to better support collaborative work for large scientific projects using visualization. Social aspects are inevitable in collaborative work. The study of social aspects often involves exploring the structure of participant roles, awareness, and trust. Furthermore, social aspects include both social and cognitive presences. “Social presence reflects the ability to connect with members of a community of learners on a personal level. Cognitive presence is the process of constructing meaning through collaborative inquiry” [51]. Social and cognitive presences are also needed for online collaboration [104].

CHAPTER 3: SPECTRUM-BASED NETWORK TOPOLOGY ANALYSIS

3.1 Introduction

In this chapter, we present a new network visualization approach for analyzing network topologies, including both the global topology structure and the importance of individual nodes and edges to each network community. In our approach, we mainly study undirected and un-weighted networks without self-loops, as they are challenging to analyze and the other network representations can be decomposed to this format. Our approach provides both an automatic network layout algorithm and interactive analysis tools for analyzing network topologies.

Our automatic graph layout algorithm produces topology-aware network visualization through utilizing the features of node distribution and coordinates in the spectral space. Specifically, network nodes are projected to a high dimensional sphere with a node dispersion algorithm, which captures the quasi-orthogonal patterns of loose communities in the spectral space. We further design interactive visualization and analysis functions to study complex networks, including topology-aware filtering, zooming, and interactive exploration. At the end of the chapter, we provide case studies and comparisons to demonstrate the advantages of spectrum-based methods.

Compared to the previous network visualization methods, our approach has two main contributions. First, the new network layout algorithm facilitates topology-aware visualization, which reveals the structure of communities in a network. Second, our network visualization provides interactive analysis functions to analyze both the global network topology and the roles of individual nodes and edges to their communities, which are previously hard-to-obtain. The efficiency of our approach is rooted from a new data transformation view based on the spectrum-based data analysis, which is significantly different from the previous force or energy driven methods.

3.2 Spectrum-based Network Layout

In this section, we first introduce the foundation – spectrum-based network analysis theorem. Then, we describe our network visualization algorithm, which consists of node projection, dispersion, and warping procedures.

3.2.1 Spectrum-based Network Analysis

A network or graph $G(V, E)$ is a set of n nodes V connected by a set of m links E, where V denotes the set of nodes and $E \subseteq V \times V$ is the set of links. G can be represented as a symmetric adjacency matrix $A = (a_{ij})_{n \times n}$. Since this chapter concentrates on analyzing data with binary adjacency matrices, $a_{ij} = 1$ if node i is connected to node j and $a_{ij} = 0$ otherwise.

Graph spectral analysis deals with the analysis of the spectra (eigenvalues and eigenvector components) of the nodes in the graph. Let λ_i be the eigenvalues of the adjacency matrix A and x_i be the corresponding eigenvectors. When $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, the spectral decomposition of A is $A = \sum_i \lambda_i x_i x_i^T$. Let x_{ij} denotes the j'th entry of x_i.

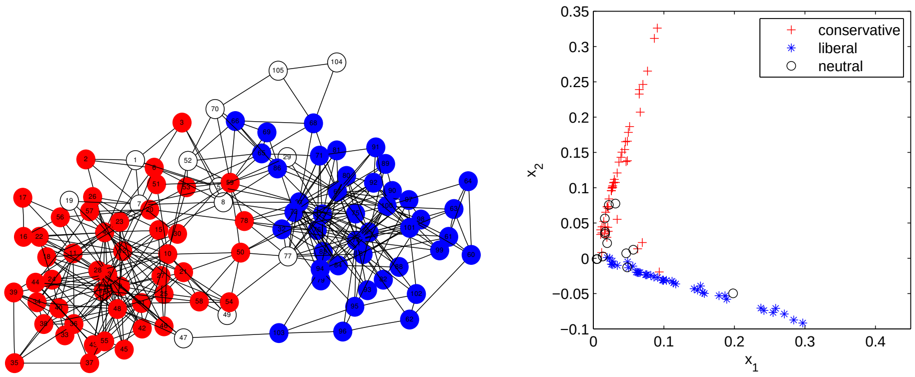
$$\begin{array}{cc}
 & \begin{array}{ccccccc} x_1 & & x_i & & x_k & & x_n \\ & & \downarrow & & & & \end{array} \\
\alpha_u \rightarrow & \left(\begin{array}{cc|c|cccc}
x_{11} & \cdots & x_{i1} & \cdots & x_{k1} & \cdots & x_{n1} \\
\vdots & & \vdots & & \vdots & & \vdots \\
\hline
x_{1u} & \cdots & x_{iu} & \cdots & x_{ku} & \cdots & x_{nu} \\
\hline
\vdots & & \vdots & & \vdots & & \vdots \\
x_{1n} & \cdots & x_{in} & \cdots & x_{kn} & \cdots & x_{nn}
\end{array}\right)
\end{array} \tag{1}$$

We can see from Formula 1 that the eigenvector x_i is represented as a column vector. The row vector $\alpha_u(x_{1u}, x_{2u}, \cdots, x_{nu})$ represents the coordinates of node u in the n-dimensional spectral space. It is known that there is an intimate relationship between the combinatorial characteristics of a graph and the algebraic properties of its adjacency matrix. Ying et al. [137] have explored the relationships of spectral coordinates and communities as follows. Here, communities are loosely defined as collections of network nodes that interact unusually frequently.

Theorem 1 (Quasi-Orthogonal Property): For a graph with k communities, the coordinate of node u in k-dimensional space, $\alpha_u = (x_{1u}, x_{2u}, \cdots, x_{ku}) \in \mathbb{R}^{1\times k}$, denotes the likelihood of node u's attachment to these k communities. Node points within one community form a line that goes through the origin in the k-dimensional space. Nodes in k communities form k quasi-orthogonal lines in the spectral space. [139]

Theorem 1 can be proved through optimizing the graph division process, which maximizes the overall edge to node densities inside each divided community. The maximum density reaches the sum of all the eigenvalues only when the quasi-orthogonal property is ensured. Please refer to [139] for details of the proof.

Noticing k is the community number of a network, which is much less than the node number of the network n. We pick a subset of the kth largest eigenvalue/eigenvector pairs. We basically ignore all the other eigenvalues and eigenvectors but the first k dimensions in the coordinate. The question of how to pick the right k will be discussed in Section 3.4.3.



(a) Polbooks network [137]

(b) Spectral space

Figure 1: Demonstration of Quasi-Orthogonal Property with the polbooks network. Two communities can be identified through the quasi-orthogonal lines in the spectral space.

We use the political book network (Valdis Krebs. http://www.orgnet.com/) as an example to demonstrate the relationship between eigenvectors and network topology. The political book network contains 105 nodes and 441 edges as shown in Figure 1(a). In this network, nodes represent books about US politics sold by the online bookseller Amazon.com while edges represent frequent co-purchasing of books by the same buyers on Amazon. Each node is labeled as "liberal" (blue), "neutral" (white), or "conservative" (red). These alignments were assigned separately by Mark Newman based on a reading of the descriptions and reviews of the books posted on Amazon.

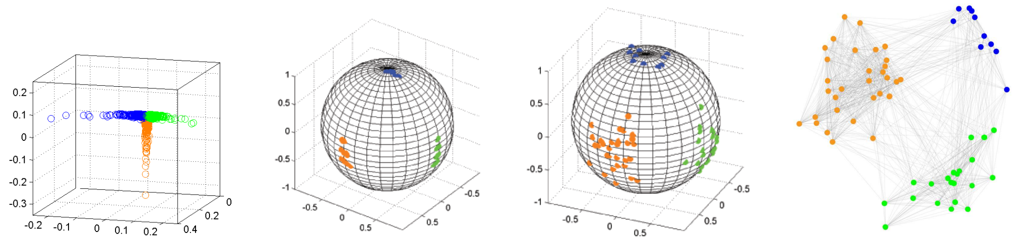
Figure 1(b) plots node coordinates projected in the 2-D spectral space of the adjacency matrix A. We can observe from Figure 1(b) that the majority vertices projected in the 2-D spectral space distribute along two straight and quasi-orthogonal lines. It indicates that there exist two communities with sparse edges connecting them. The first up-trend line consists of most nodes in red while the second down-trend line consists of most nodes in blue. White nodes, which correspond to either noise nodes or bridging nodes, distribute either around the origin or between two quasi-orthogonal lines in the projected space.

3.2.2 Spectrum-based Network Layout Algorithm

Based on the quasi-orthogonal property (Theorem 1), we have designed a network layout method that visualizes the network topology, especially for exploring the structure of network communities and the node relationships. Our basic idea is to project nodes onto a k-dimensional sphere in the spectral space, as it preserves main structure of the network topology and maximizes the distances between different network communities. Later, we visualize the network spectrum relationships on the k-dimensional sphere surface through a 2-D plane, where allows convenient interactive analysis. As shown in Figure 2, our approach consists of the following three steps: node projection, node dispersion, and sphere warping.

3.2.2.1 Node Projection in the Spectral Space

The first step is to project nodes to a high-dimensional sphere in the spectral space. Given an adjacency matrix A and a user-assigned parameter k, we calculate the eigenvalues and eigenvectors x_i of matrix A. As shown in Formula 1, the coordinates



(a) Quasi-orthogonal property (b) Node Projection (c) Node Dispersion (d) Sphere Warping

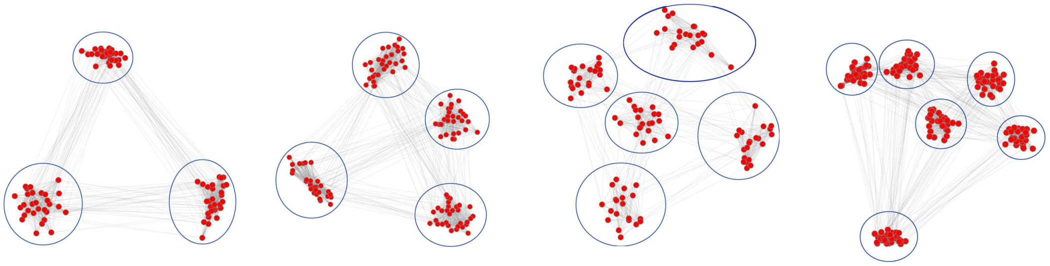
Figure 2: The pipeline of our algorithm. (a) Quasi-orthogonal property of the node distribution in the spectral space. (b) In the spectral space, nodes are projected onto the surface of a k-dimensional sphere. (c) A process of node dispersion in the spectral space is added to reduce the issue of node overlapping. (d) The nodes are transferred from k-dimensional sphere surface to a 2-D space for visualization and interaction. In the final layout, nodes placed near the center of each community are often more important to the network, as they have larger numbers of connections in the community than other nodes.

of node u in the first k-dimensional spectral space is the row vector $(x_{1u}, x_{2u}, \cdots, x_{ku})$.

$$[htb]x'_{iu} = \frac{x_{iu}}{\sqrt{\sum_{j=1}^{k} x_{ju}^2}} \times S_{radius}, 1 \leq i \leq k. \tag{2}$$

Then, we project the nodes onto a k-dimensional sphere whose center is located at the origin of the spectral space. In Equation 14, S_{radius} is the radius of the sphere. In all the results of this chapter, we use the value 1 for the sphere radius, as it does not affect the measure on geodesics on the k-sphere surface.

Figures 3 (a-d) show the projection results of several synthetic datasets. These datasets contain different numbers of communities, ranging from 3 to 6. Inside each community, the nodes are fully connected; while extra 10% edges are added to connect nodes from different communities. In all these four results, the values of k are the same as the number of communities. The results demonstrate that our approach can reveal the network topology from datasets with different numbers of communities.



(a) 3 communities (b) 4 communities (c) 5 communities (d) 6 communities

Figure 3: (a-d) Our network layout algorithm successfully reveals the global topology from network data with different numbers of communities. As the circles show in all these four networks, the community structures are clearly revealed.

3.2.2.2 Node Dispersion in the Spectral Space

As shown in Figure 2(b), after all the network nodes are projected onto the k-dimensional sphere, we may have communities with clustered nodes when they are densely connected. Generally, users prefer to spread the nodes in the network layout, so that they can not only visualize the main topology structure of the network, but also analyze the connections of individual nodes and edges. Therefore, our second step is to disperse nodes around the k-dimensional sphere surface in the spectral space.

Our node dispersion method is designed based on the following property of node coordinates in the spectral space from [139]:

Property 1 (Distance): The distance of a node to the origin of the spectral space indicates the degree of the importance of this node to its community. The farther a node is away from the origin, the more important the node is in its community.

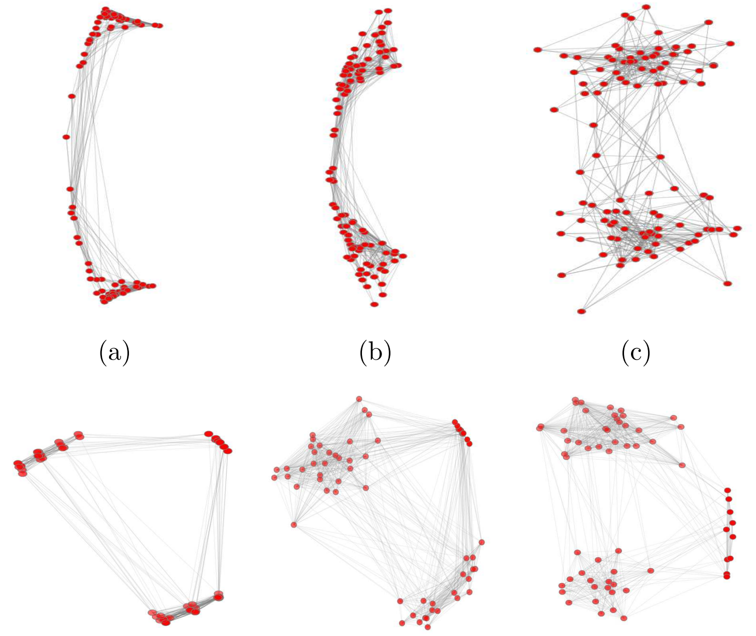
According to the distance property, nodes closer to the origin of the spectral space are more likely to be random nodes, which do not belong to any communities in the network obviously; while nodes away from the origin are often crucial to their communities [139]. Therefore, our design principle is to preserve the locations of nodes away from the origin of the spectral space and disperse all nodes on the sphere surface according to their distances to the origin.

We disperse the nodes around their projection locations on the k-dimensional sphere, instead of their original coordinates in the spectral space. Since a small amount of movements around the original coordinates can result in a large change on the projection locations, especially when a node is close to the origin, our approach moves each node randomly around the sphere surface according to its distance to the origin of the spectral space. In this way, the center of every community, represented by the intersection of the corresponding quasi-orthogonal line and the k-dimensional sphere, remains on the same location. Since important nodes with large connections are distributed near the surface of k-dimensional sphere, they are located close to their projection locations. On the contrary, random nodes, distributed nearer the center of the spectral space, are more affected by noises.

As shown in Equation 3, $distance_u$ is the distance of node u to the origin; r is a random number from -1 to 1; and R is a parameter to control the degree of dispersion. The selection of R is dependent on the number of nodes in the network. Generally, we use a small value for large networks to avoid nodes that are previously separated from different communities interweaving on the sphere after the node dispersion step. While it is ideal to move the nodes on the sphere surface, the computation involves calculating the angles for each of the k dimensions, which is more time-consuming. Our approach is computationally efficient and achieves similar dispersion effects.

$$x''_{iu} = x'_{iu} + R \times (1.0 - distance_u) \times r, 1 \leq i \leq k. \tag{3}$$

As shown in Figure 4, we find that sometimes it is still hard to disperse nodes on networks with dense nodes. Figure 4(b) shows that the nodes spread out mainly in one dimension, instead of the entire 2-D space. Therefore, we extend the k dimensional coordinates to $k+1$ dimensions with $x_{(k+1)u} = 0$, which introduces an additional dimension to disperse the nodes. Then, we perform the above node dispersion method on the $k+1$ dimensions. The result in Figure 4(c) shows that this step successfully disperses the nodes on the 2D space.



(a) (b) (c)

(d) Direct projection (e) k dispersion (f) $k+1$ dispersion

Figure 4: The results of the node dispersion on the polybook dataset (a-c) and a synthetic dataset with three communities, including 60 nodes and 871 edges (d-f). (a) and (d) show the layouts without node dispersion. (b) and (e) show the layouts with node dispersion in k-dimensional space. (c) and (f) show the layouts with node dispersion in $k+1$-dimensional space, which produce the best network layouts.

3.2.2.3 Sphere Warping to 2D Network Layout

The final network layout is generated by warping the surface of a k-dimensional sphere to a 2D space. To achieve this effect, we first collect a distance matrix, which measures node distances on the surface of the k-dimensional sphere. Specifically, for any two nodes u and v, we calculate the angle θ between the vectors $\overrightarrow{Ou}$ and $\overrightarrow{Ov}$, where $\overrightarrow{O}$ represents the origin of the spectral space. As $\overrightarrow{O}$ is a k-dimensional zero vector, we can calculate the angle according to Equation 4:

$$\theta = arccos(\frac{x''_{iu}x''_{ju}}{|x''_{iu}| \times |x''_{ju}|}) \tag{4}$$

Then, the distance of two nodes on the surface of the k-dimensional sphere can be calculated via Equation 5. As the distances ($spherical - distance$) are linear to the angles θ, we can actually ignore this calculation and achieve the same network layout effects.

$$spherical - distance = \frac{\theta \times \pi \times S_{radius}}{180^\circ} \tag{5}$$

At the end, the problem is how to generate a 2D network layout, which can best approximate the node pair distances we measure from the sphere surface. Among different high-dimensional projection methods, we choose the multi-dimensional scaling (MDS) mechanism, as it optimizes the preservation of the relative node pair distances from the k-dimensional sphere to the 2D layout. As all the results in Figures 3 and 4 show, nodes that belong to the same community are projected onto similar locations on the k-dimensional sphere and grouped nicely on the final network layout.

3.3 Spectrum-based Visualization and Interaction

We further develop a network visualization approach to analyze the roles of individual nodes and edges to the global topology structure. This is achieved by incorporating a framework of spectrum-based non-randomness measurements. We have also designed interaction methods that allow users to select and analyze their interested nodes or regions according to the network topology.

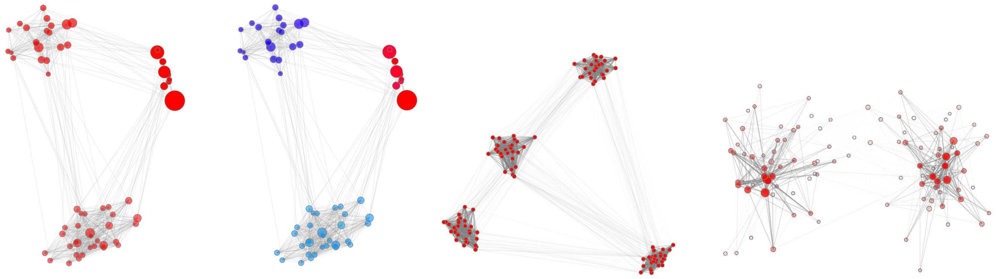
3.3.1 Spectrum-based Network Visualization

We first introduce the non-randomness framework, which is used to visualize the importance of a node or a link to the network topology. For example, an individual's social network tends to consist of members of the same ethnic group, race, or social class. Intuitively, two friends of a given individual are more likely to be friends with each other than they are with other randomly chosen members. The non-randomness framework has been recently presented by Ying et al. [139] [137]. It quantifies all graph non-randomness measures mathematically from the spectra of the adjacency matrix of the network. The framework begins with a study of edge non-randomness by spectral coordinates of its two connected nodes in the spectral space. The node non-randomness is then defined as the sum of non-randomness values of all edges that connect to it. The formal result is given below.

Denote $\alpha_u = (x_{1u}, x_{2u}, \ldots, x_{ku}) \in \mathbb{R}^k$ as the coordinates of node u and $\alpha_v = (x_{1v}, x_{2v}, \ldots, x_{kv}) \in \mathbb{R}^k$ as the coordinates of node v in the spectral space.

Property 2 (Edge Non-randomness): The edge non-randomness $R(u, v)$ is defined as $R(u, v) = \alpha_u \alpha_v^T = \sum_{i=1}^{k} x_{iu} x_{iv}$. We then have $R(u, v) = \|\alpha_u\|_2 \|\alpha_v\|_2 \cos(\alpha_u, \alpha_v)$.

Property 3 (Node Non-randomness): The node non-randomness $R(u)$ is defined as $R(u) = \sum_{v \in \Gamma(u)} R(u,v)$, where $\Gamma(u)$ denotes the neighbor set of node u. We then have $R(u) = \sum_{i=1}^{k} \lambda_i x_{iu}^2 = \alpha_u \Lambda_k \alpha_u^T$, where $\Lambda_k = diag\{\lambda_1, \lambda_2, \ldots, \lambda_k\}$, which means the non-randomness of node u is the length of its spectral vector with the eigenvalue weighted on corresponding dimensions.



(a) Node transparency, size and color (b) Distance-based color (c) Edge transparency (d) Visualization of the important nodes and edges

Figure 5: Example results of our approach. (a-c) demonstrate results of two synthetic datasets: (a) and (b) use the synthetic dataset with three communities, including 60 nodes and 737 edges; (c) is the synthetic dataset with four communities, including 120 nodes and 2262 edges. We also demonstrate the visualization strategies on real Polbook dataset (d). The layout clearly visualizes the network community structure and emphasizes the crucial nodes with their connections in the dataset.

The properties of nodes and edges in the spectral space can be used to reveal important network topology. For example, we can color the nodes according to their distances to the origin of the spectral space. With the HSV color model, we adjust the hue channel with normalized distances and set the other two channels to be full for the most visible effect. It is very interesting to see that this coloring scheme automatically visualizes the three communities differently, as shown in Figure 5(b). This effect is resulted when the communities are located at different distances to the origin in the spectral space. Similarly, we design our network visualization based upon the non-randomness properties to emphasize the important network nodes and edges.

Node transparency (demonstrated in Figures 5(a & d)): According to Property 1 (distance), the farther a node is away from the origin, the more important the node is in its community. Therefore, we set the transparency of a node to be linear to its distance from the origin.

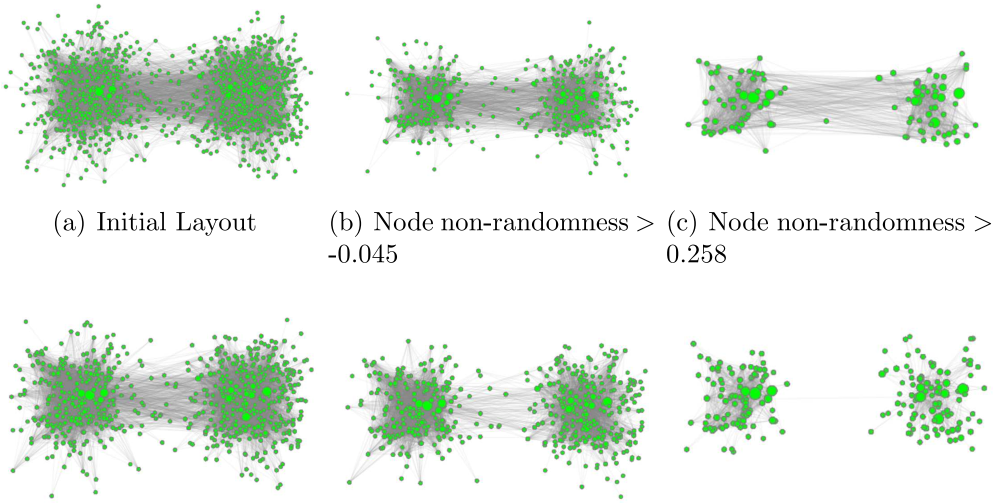
Node size (demonstrated in Figures 5(a & d)): According to Property 3 (node non-randomness), we set the size of a node to be linear to its non-randomness value. All the node non-randomness values are normalized before the visualization.

Node color: Generally, we preserve this rendering parameter to visualize additional attribute of the network data. As Figure 5(a) shows, we just use one color for all the nodes.

Edge transparency: According to Property 2 (edge non-randomness), we set the transparency of an edge to change linearly with its non-randomness value. All the edge non-randomness values are normalized to 0-0.5 before the visualization. As Figures 5 (c and d) show, the important edges are often inside each community.

3.3.2 Interactive Topology Analysis

We present three interactive analysis methods which allow users to select or filter network data to analyze network topology and the roles of individual nodes and edges. The interactive analysis methods can be combined freely during interactive exploration, which is often necessary for understanding complex networks.



(a) Initial Layout

(b) Node non-randomness > -0.045

(c) Node non-randomness > 0.258

(d) Edge non-randomness > 0.001

(e) Edge non-randomness > 0.003

(f) Edge non-randomness > 0.01

Figure 6: Node and edge filtering results on PolBlogs dataset. The node filtering process is demonstrated on the top (b & c) and the edge filtering process on the bottom (d & f). The range of node non-randomness on PolBlogs is from -0.3495 to 5.7343 and the range of edge non-randomness is from -0.0016 to 0.0314. With more node and edge filtering, the kernel topology structure appears clearer.

3.3.2.1 Topology Analysis through Filtering

For large networks, the main topology structure is often hidden in the visualization of all the nodes and edges. Filtering in our interaction, not just according to data attributes, provides a tool to remove irrelevant nodes and edges and visualizes the main network topology. According to the Properties 2 and 3, the nodes and edges with low non-randomness values are random nodes and edges, indicating that they do not possess obvious connection relationships to any major communities in the network. Therefore, we can use the non-randomness measurements to filter nodes and edges in the visualization.

The main benefit of this method is that we can select nodes along the main topology structure, ranging from visualizing only the kernel nodes of each community to all the nodes in the network. As items in Figures 6 show, (a) visualizes all the network nodes and (c) reveals only the most important nodes. The same selection process can be applied to the edges for revealing important edges to the network topology, as shown in Figures 6 (d,e and f).

3.3.2.2 Topology Analysis through Selecting Examples

For complex network structures, we provide an interaction method to help users adjust the network layout based on their knowledge. The approach allows users to identify communities through selecting representative nodes and simplify the network by removing the known communities. The interactive selection process can be iterated until the network topology is completely unraveled. Generally, with a new network, we start from $k = 2$ and increase the value gradually. We can also identify the representative node of a community by selecting the node with the maximum connection number in the community.

During and after the interactive analysis process, we visualize the network with our network layout algorithm using k equals to the number of representative nodes. We have designed an automatic algorithm to relocate the other nodes surrounding the representative nodes. Assume that m representative nodes are $(v_1, v_2, \ldots, v_m)$. For any other node u, we first determine which of the m communities that the node u belongs to with a clustering algorithm. In this chapter, we use k-means clustering algorithm, as it is widely-used and allows us to define the number of communities/clusters freely. Second, a node u is shifted from its original location in the spectral space as follows, where p is a parameter to control the amount of movement.

$$\vec{u}' = \vec{u} + p \times \frac{\vec{uv_i}}{\|\vec{uv_i}\|} \tag{6}$$

As Figures 8 and 9 show, this achieves the effect that the all the rest nodes in each community are close to each selected community representative node. This interaction function is especially useful to analyze complex networks.

3.3.2.3 Detail Observation through Zooming

The third interaction method is the zoom in/out function, which enables users to visualize the details of a selected region in the network. We allow users to select the center of one region which they are interested in. Then, with the x and y axis values of the selected center, the zoom-in region is $x - \delta < x' < x + \delta, y - \delta < y' < y + \delta$, δ can be adjusted by users, that can help them to see the different levels of zoom-in results. As Figure 7 shows, the initial network visualization may contain too many nodes with some overlapping. The zoom-in layout shows more details compared to the selected region marked in black.

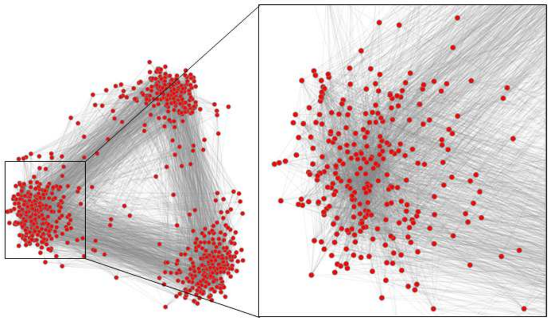
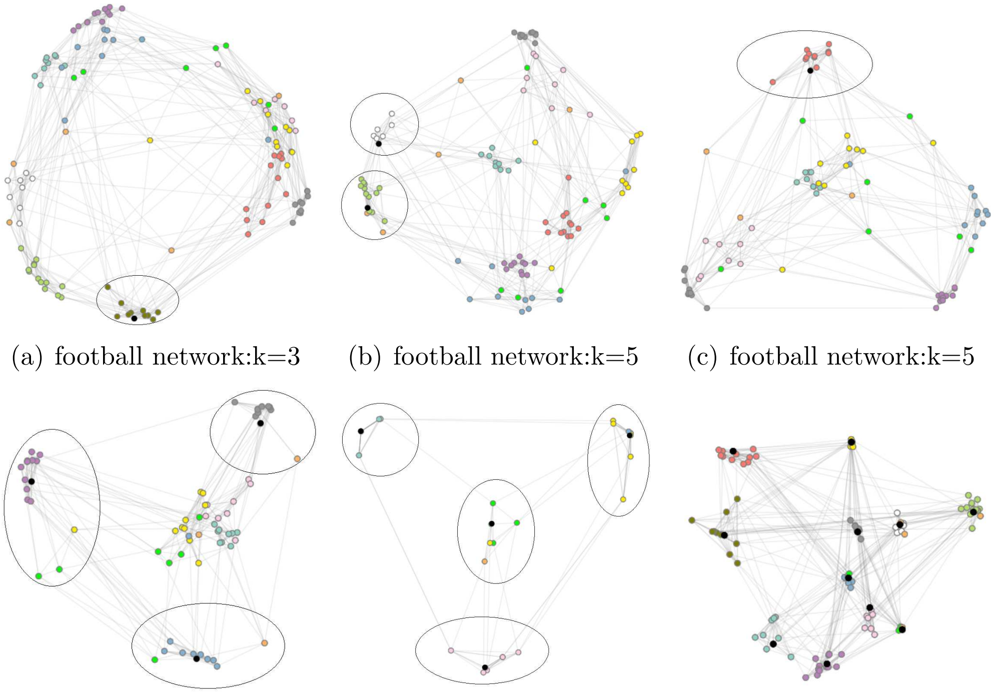


Figure 7: The zooming result of a synthetic dataset with 3 communities, including 300, 250 and 230 nodes respectively. The zooming function allows more details to be visualized.

3.4 Results and Discussion

3.4.1 Results



(a) football network:k=3

(b) football network:k=5

(c) football network:k=5

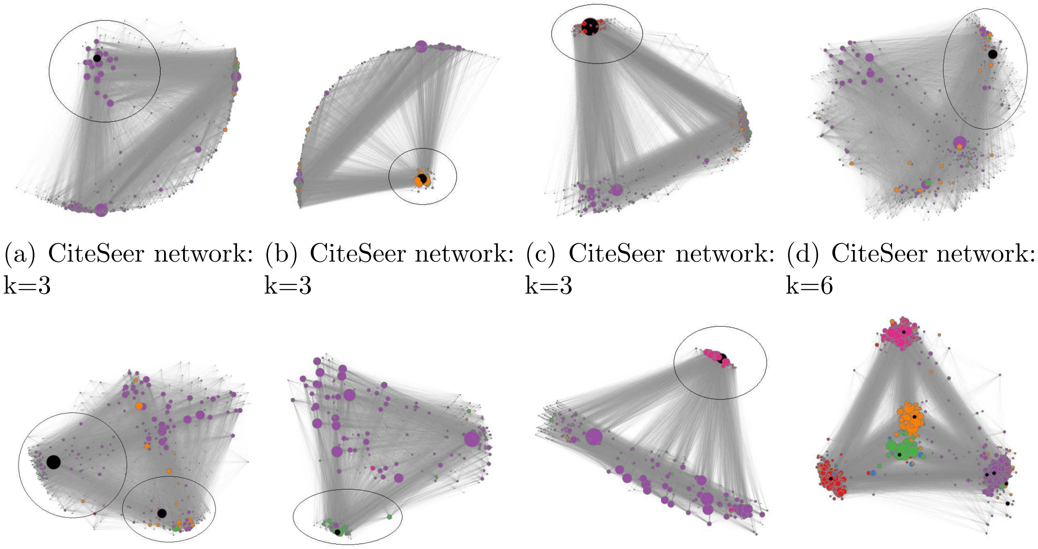
(d) football network:k=7

(e) football network:k=4

(f) Final layout of football network

Figure 8: Examples of interactive analysis for football datasets. Figures (a-e) and (g-m) demonstrate the selection of communities (each circled separately) and representative nodes (rendered in black). The value of k for each step is labeled under each figure. Figure (f) show our final results of these two real datasets, which match the topology structure of original datasets closely.

We have tested our approach with both real and synthetic network datasets. For example, Figures 6 visualize the PolBlogs data, which contains 1222 nodes and 33428 edges. Figure 5(d) visualizes the PolBook data, which contains 105 nodes and 441 edges. The details of each dataset are provided in the descriptions of the figures. Other figures use synthetic datasets with different numbers of communities and noise levels.



(a) CiteSeer network: k=3
(b) CiteSeer network: k=3
(c) CiteSeer network: k=3
(d) CiteSeer network: k=6
(e) CiteSeer network: k=5
(f) CiteSeer network: k=5
(g) CiteSeer network: k=3
(h) final layout of CiteSeer network

Figure 9: Examples of interactive analysis for CiteSeer dataset. Figures (a-h) demonstrate the selection of communities (each circled separately) and representative nodes (rendered in black). The value of k for each step is labeled under each figure. Figure (h) shows our final results of these two real datasets, which match the topology structure of original datasets closely.

The following demonstrates the interactive analysis with two real datasets. We compare our results with the original topology structures described in the data and provide discussion on how our network visualizations reveal meaningful information in the datasets. To demonstrate our interactive analysis approach, we render the nodes according to the community structure so that all the nodes in one community have the same colors.

The first dataset is the network of American College football games between Division IA colleges during regular season Fall 2000. There are 115 teams and 12 communities, including one for independent teams. As shown in Figure 8 (a), initially the network topology can separate several large communities; however, there are nodes obviously floating between communities. We start to choose tightly grouped communities and representative nodes, shown in Figures 8 (a-e). Every time we start from $k = 2$ and gradually increase the value of k, so that we can choose clearly isolated communities. We stop at Figure 8 (e), since the topology structure is very clear. According to the selected 11 representative nodes, we set $k = 11$ and generate the final network layout, shown in Figure 8 (f).

Comparing our result with the original network topology, we can tell that major topology structures are identified successfully. The main difference is caused by the independent conference, rendered in orange. The teams in this conference have more connections with other conferences than their own conference. The teams in the other 11 communities are all categorized correctly.

The second dataset is the CiteSeer dataset from the paper "Collective Classification in Network Data", which includes 4536 edges and 3312 scientific publications from 6 research communities. Figures 9 (a-h) demonstrate the interactive analysis process for selecting groups and representative nodes. Node filtering and edge filtering are applied to visualize important topology structure. The sizes of nodes are also changed with the distance of the node to the origin in the k dimensional spectral space.

Since the purple community ("Information Retrieval") is the largest in the network (containing more than 30% connections than any other groups), the initial networks are dominated by the purple group and four representative nodes are selected from this group during the interactive process. This does not affect the final layout, where the four purple representative nodes are closely located and this group is still clearly separated.

Figure 9 (h) visualizes the final layout of the citation network with five separated communities, purple (IR), red (Agents), green (DB), orange (ML), and pink (HCI). The only missing community, blue (AI), is overlapped with the red (Agents) group and dragged to the direction towards the green (DB) and orange (ML) communities. This is caused by the fact that the connections between communities Agents and AI are twice the connections inside the community AI. Some blue nodes locate close to the orange (ML) and green communities (DB), as the AI community has more connections with these two community members than the pink (HCI) and purple (IR) communities.

Note that the purpose of interactive analysis is mainly for simplifying complex network and choosing representative nodes. The approach can tolerate errors in the selection process. For example, all the three communities selected in Figure 8 (d) include nodes from other communities. They are corrected in the final result shown in Figure 8 (f), when the relationship of each node is recomputed globally.

3.4.2 Computation Complexity and Scalability

The computational complexity of our approach is $O(N^2)$, which is relatively fast. Table 1 shows the time complexity of each step in our approach, where N is the number of nodes in the network. The first step, spectral decomposition, is performed with QR algorithm by a reduction to Hessenberg form (please refer details to Lloyd N. Trefethen and David Bau, Numerical Linear Algebra, 1997). The third step, with MDS, follows the hybrid approach from Information Visualization in 2003 (please refer details to “Fast Multidimensional Scaling through Sampling, Springs and Interpolation”). Active researches are performed to accelerate spectral decomposition and

MDS, which can further used to improve the performance of our approach.

The scalability of our approach is mainly determined by the approaches of spectral decomposition and MDS. Both approaches have been shown to handle large-scale data, as they are common problems shared among the communities of math, data mining, graph drawing, information visualization, and bio-informatics, etc. The second step is only linear to N, which can be easily performed and accelerated. We also provide interactive visualization functions to analyze complex networks, which assist users to filter random nodes and explore main topology structures.

Table 1: Time complexity of our approach.

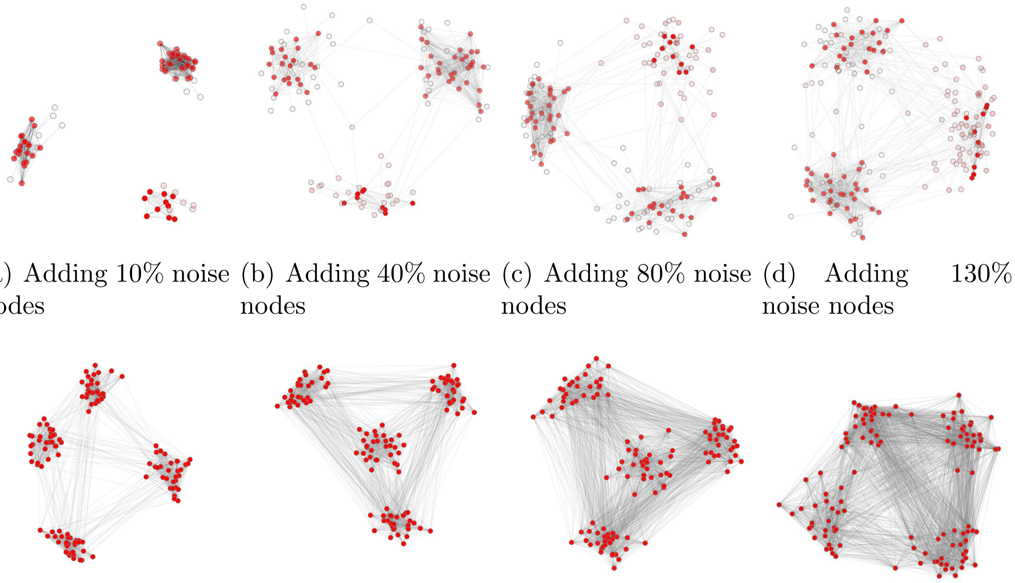
Steps	Computation Complexity
Spectral Decomposition	$O(N^2)$
Node Projection and Dispersion	$O(N)$
Space Warping	$O(N^2)$

3.4.3 Resistance to Noise

Figures 10 demonstrate that our network layout algorithm can successfully reveal the global topology structure when a large amount of noises is introduced into the network. Two synthetic datasets with three and four fully connected communities are generated: one containing three communities with 60 nodes and 670 edges; the other containing four communities with 120 nodes and 1740 edges. In order to simulate the real cases, we add random noise nodes and noise edges to the datasets. Specifically, 10, 40, 80 and 130 percentages of noise nodes are introduced respectively and connected to the existing nodes randomly. Noise edges (10, 30, 80 and 120 percentages of total edges) are introduced by randomly adding an edge between nodes from different communities. The noises disrupt the graph layout and enrich complexity of the network data.

In Figures 10, the transparency of the nodes are measured according to the distance of a node from the origin of spectral space. After observing all these results, we find that almost all the bright red nodes are located inside the communities while noise nodes are located outside sparsely. With our node filtering interaction, users can easily identify the main network topology. Figures 10 (a-d) demonstrate the results with different numbers of noise nodes. The main network topology structures can be visualized successfully even adding 130% of the noise nodes. In this case, the original nodes in the network are better representatives of their communities, as they have more connections than noise nodes. Our approach successfully visualize the original nodes as bright red nodes which are located inside the communities. The noise nodes are located outside sparsely and rendered with transparent colors. Thus, based on Property 3, the node non-randomness of these representative nodes should be larger. We provide a visualization strategy in section 4.1, which allows users to emphasize these representative nodes using the node non-randomness. Also, with our node filtering interaction, users can easily identify the main network topology.

Figures 10 (e-h) demonstrate the results with different numbers of noise edges. We also use the same coloring scheme on this dataset. Since all the nodes in the network are important to a community, their colors are calculated to be bright. In this case, with the increasing of the noise, the four network communities become closer. Our approach can reveal the network topology structure even adding 120% of the noise edges.



(a) Adding 10% noise nodes

(b) Adding 40% noise nodes

(c) Adding 80% noise nodes

(d) Adding 130% noise nodes

(e) Adding 10% noise edges

(f) Adding 30% noise edges

(g) Adding 80% noise edges

(h) Adding 120% noise edges

Figure 10: (a-d) Adding noise nodes to a network with three communities. (e-h) Adding noise edges to a network with four communities.

3.4.4 Choice of k

The parameter k controls the number of effective dimensions in the spectral space. As Figures 11 show, the parameter k in our approach is important to the network layout. This result is demonstrated with a synthetic dataset, which has 3 communities with 300, 250 and 230 nodes respectively. As shown in Figures 11, the result of $k = 2$ suggests two clusters at the left and right ends, but there are strong correlations between the two clusters. The result of $k = 3$ produces the best result in the sense that all the nodes are obviously clustered to three communities and there are only several nodes between each pair of communities. From the results of $k = 4$ and $k = 5$, the visualizations still suggest a topology structure of three communities. Only the node distributions are not as clear as the result of $k = 3$. This is caused by the mixture of the main topology structure and small details inside the three communities. Even without pre-knowledge of this dataset, we can determine that there are three communities in the network, as the result with $k = 3 = \#communities$ brings the best network layout with clear node clusters.

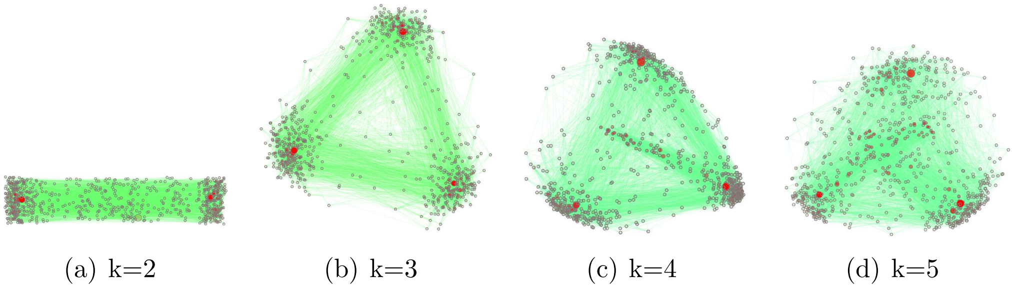
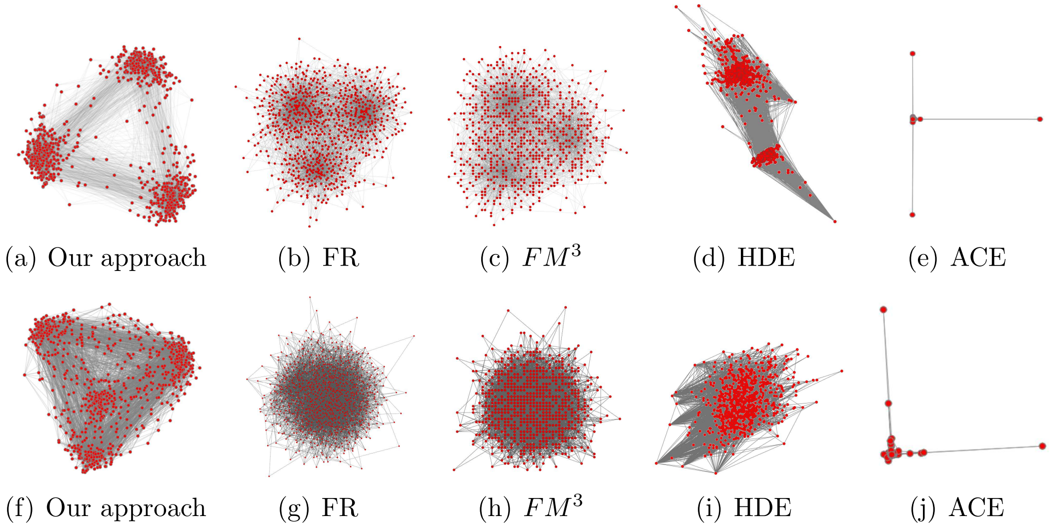

(a) k=2 (b) k=3 (c) k=4 (d) k=5

Figure 11: Demonstration of the effects of parameter k on the network layouts.

Generally speaking, the value of k should be equal to the number of the communities in the main topology structure, as it defines the dimensions used in the layout algorithm. If k is small, some important information in higher dimensions are not considered in the network layout algorithm. On the other hand, if k is large, small details inside each community can affect the main topology structure. Up to now, the choice of k is still an open problem in the fields of data mining, statistics and visualization. No single algorithms can successfully compute the best value of k. Therefore, in this chapter, we design interactive analysis methods, allowing users to select a representative node for each community and utilizing this knowledge to adjust the network layout. With this data transformation, we can not only tell the number of communities, but also the relationships of each node to the community.

3.4.5 Comparison

We have compared our results with four network visualization approaches. The layouts of Fruchterman and Reingold algorithm, FM^3 and HDE are generated by Tulip graph visualization system (http://tulip.labri.fr/TulipDrupal). The ACE layouts are generated by the original Yehuda Koren's code and provided by Dr. Daniel Archambault.


(a) Our approach (b) FR (c) FM^3 (d) HDE (e) ACE

(f) Our approach (g) FR (h) FM^3 (i) HDE (j) ACE

Figure 12: Comparison results: the top row visualizes a dataset including 3 communities with 300, 250 and 230 nodes respectively. The number of edges between communities is approximately 20% of the edges inside all the communities. The bottom row visualizes a dataset containing 4 communities with 200, 180, 170 and 290 nodes respectively. The number of edges between communities is approximately 72% of the edges inside all the communities. Our approach achieves the best results by correctly separating the communities and clearly visualizing the topology structures.

Fruchterman and Reingold algorithm (FR): The FR algorithm is a classical force-directed approach [47], which has been very widely used in network visualization. The network is simulated as a physical system where nodes are pulled closer or pushed further by the assigned edges.

As shown in Figure 12, the FR algorithm visualizes the topology of three communities with just 100 iterations for the first dataset (b). Comparing to (a), our result achieves a clearer visualization of the community structure, as the communities are well separated. We have also tried the FR algorithm with other numbers of iterations, the results are almost the same even at 10000 iterations. For the second dataset, whose community structure is less obvious (ratio of edges between communities to inside communities is higher), the FR algorithm fails even at 10000 iterations (g); while our approach still successfully reveals the topology of 4 communities.

Comparing with the FR algorithm, our approach has two advantages. First, our approach is computationally faster with a complexity of $O(N^2)$, while the FR algorithm is $\Theta(N^2 + E) \times I$, where I is the number of iterations. Second, our approach is non-iterative, while the FR requires users to adjust the number of iterations, which can affect the network layout results.

Fast Multipole Multilevel Method (FM^3): The FM^3 algorithm is an accelerated force-directed approach [62], which comes with an efficient multilevel scheme and a force model. The time complexity of FM^3 is $O(N \log N + E)$, which is faster than our approach.

Figure 12 (c) demonstrates the FM^3 algorithm for the first dataset with 30 iterations. The results remain the same with larger numbers of iterations. While this result suggests a topology of 3 communities, it is not as clear as the layouts in (a) and (b). For the second dataset, as shown in Figure 12 (h) (10000 iterations), the FM^3 algorithm only generates layout with evenly distributed nodes without clear topology structures. This indicates that the FM^3 algorithm does not work well for densely connected networks.

High-Dimensional Embedding (HDE): The HDE algorithm [64] is an elegant spectral approach, which is the closest algorithm to our method. This algorithm generates graph layouts through embedding information from a network to a high dimensional space and projecting the nodes to a 2D plane.

The HDE algorithm works well on networks with mesh-like structures, however it does not generate correct layouts for our test datasets. As shown in Figure 12 (d), the bottom group only includes one node, which is not correct. Figure 12 (i) does not show the correct community structure.

Algebraic Multigrid Optimization (ACE): The ACE algorithm [83] is described in our related work. It is an extremely fast algorithm, which is especially suitable for large graphs. Similar to HDE algorithms, ACE is a spectral approach involving of computing node projections in multi-dimensions.

As shown in Figures 12 (e) and (j), most nodes gather together in one group, with only 3 nodes outside (e and f). The performance of the ACE algorithm is the fastest among the five approaches; however, the comparison results indicate that the ACE algorithm does not work well for our densely connected networks.

The results above compare automatic network layout algorithms. With the increasing size and complexity of networks, we believe that interactive analysis should also be included. For example, filtering techniques remove random nodes and edges to reveal main network topology. Techniques of interactive network visualization have also been explored to extract or modify network hierarchy, which can handle large-scale datasets through incorporating expert knowledge.

3.5 Conclusion

This chapter presents a consistent network visualization approach through integrating advanced spectrum-based network analysis techniques. We have demonstrated the advantages of our approach with several real and synthetic datasets in the case studies and comparisons. Our results capture many useful network topology clues hidden in the spectral space, allowing effective analysis of network topology. This data transformation framework is different from previous network layout approaches, which often come directly from the network connection information. This work is also different from previous spectrum-based approaches, which often concentrate on showing the data with common interests.

Our approach can be used to analyze general network data, especially for social networks. Popular social networks, such as Facebook and LinkedIn, have become part of our daily lives and made enormous impacts on the society. Our approach is designed to study community structures in such networks with both automatic graph layout and interactive analysis functions. The computation complexity and scalability of our approach demonstrate potential to handle large-scale networks. Our approach can be also used for other networks. For example, frauds or attacks can be identified for security since malicious nodes often demonstrate different distribute patterns in the spectral space. We believe that advanced spectrum-based analysis techniques should be explored for developing more effective data transformation and visual analytics approaches.

CHAPTER 4: COMMUNITY EXPLORATION FOR SIGNED NETWORKS

4.1 Introduction

Social networks are changing our everyday lives. Previous network visualization researches have concentrated on networks with only positive edges [81]. However, many social networks in real-life are signed networks, which can reflect a wide range of relationships, such as like/dislike and friend/enemy. While ideas from existing techniques can be extended to signed networks, such as the concepts of force-directed approaches [80], impacts of negative edges should be studied and incorporated in the visual analysis of signed networks [81].

In this chapter, we study spectral analysis methods for revealing important relationships from signed networks. Spectrum-based methods consider matrices associated with a given network and compute their eigenvalues and eigenvectors [23]. It is known that there is an intimate relationship between the combinatorial characteristics of a graph and the algebraic properties of its adjacency matrix. However, it is often not clear how to analyze a complex network visually with spectral analysis theories, as they are abstract and nonintuitive. In this chapter, we concentrate on addressing two research challenges of visual analytics: what are the important spectral patterns and how to use them to study signed networks. Our work is built upon a theoretical foundation of spectral network analysis to studies spectral features of community structures, which is in chapter 3. Different from other spectral methods, our approach derives from empirical analysis of signed networks and provides a mechanism for studying general signed networks.

Specifically, we provide a spectral analysis framework for analyzing community structures of signed networks. We start from exploring relationships of spectral decomposition and community structures for two signature signed networks, representing the internal and external relationships of communities respectively. We have found that both signed k-block and k-partite networks demonstrate significant patterns in the k-dimensional spectral subspace spanned by the eigenvectors corresponding to the largest absolutes of eigenvalues. These results are summarized and proved formally. We also describe the spectral features of general signed networks and demonstrate that our spectral framework can be applied to analyze general signed networks.

To enable interactive exploration of signed networks, we present signed network layout and interaction approaches driven by the objective of studying community structures. The methods concentrate on essential tasks of network analysis, including finding the number of communities k and sub-communities. They provide essential information and a set of interaction mechanisms for studying community structures from various signed networks.

The advantages of the spectral analysis methods are the robustness and efficiency on capturing features of global data distributions. The main contribution of this chapter is the combination of theoretical spectral analysis and practical visualization methods to study general signed networks. Both real-life and synthetic networks are used to demonstrate the effectiveness of our approach.

4.2 Spectral Analysis of Signed Networks

The focus of our spectral analysis is to provide two crucial information to visualization: Explore signature spectral patterns that correspond to important community structures;

Provide guidelines for visual exploration of general signed networks by identifying important spectral spaces.

This section presents our spectral analysis framework to achieve the two objectives. We start with the results of our experiment to explore various spectral patterns. Formal spectral analysis are conducted on two signature signed networks, the k-block and k-partite networks, which can serve as the basic cases of general signed networks. At the end of this section, we describe how the results from the two signature networks can be extended to explore general signed networks.

4.2.1 Notations

A network or graph $G(V, E)$ is a set of n nodes V connected by a set of m links E, where V denotes the set of nodes and $E \subseteq V \times V$ is the set of links. G can be represented as a symmetric adjacency matrix $A = (a_{ij})_{n\times n}$. The links of a signed graph can be simplified as follows:

$a_{ij} = 1$ when node i and j have a positive relationship;

$a_{ij} = -1$ when node i and j have a negative relationship;

$a_{ij} = 0$ when node i and j have no relationship.

Graph spectral analysis [23] deals with the analysis of the spectra (eigenvalues and eigenvector components) of the nodes in the graph. Let λ_i be the eigenvalues of the adjacency matrix A and x_i be the corresponding eigenvectors. When $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, the spectral decomposition of A is $A = \sum_i \lambda_i x_i x_i^T$.

As chapter 3 mentioned, let x_{ij} denotes the j'th entry of x_i. As shown in Formula 1, the eigenvector x_i is represented as a column vector. The row vector $\alpha_u(x_{1u}, x_{2u}, \cdots, x_{nu})$ represents the coordinates of node u in the n-dimensional spectral space.

Ying et al. [137] have proved a theorem of quasi-orthogonality (Theorem 1 in chapter 3). Our study in this chapter extends previous spectral analysis results for unsigned networks to signed networks.

4.2.2 Experiment of Two Signature Signed Networks

Our study starts with two signature signed networks, the k-block network with only internal edges and the k-partite network with only external edges. Since all the edges of a network, no matter their signs, can only be divided to internal or external categories; the two special signed networks represent the most important community structures of a signed network. We show later how they can be used to analyze general signed networks.

In this section, based on the previous spectral theory for unsigned networks, we study the changes of eigen-decomposition and network layouts in the spectral space for signed network with negative edges. In our experiment, we use four synthetic networks, which have similar structures with 3 communities. Each community has 100 nodes. The differences of these four networks are the connection density of each community, which range from sparse-connected to extreme densely-connected community structure. Additionally, a small number of external positive edges are added for each network as noise to simulate the network connections in reality.

4.2.2.1 The Generation of Signed Networks

In this section, we introduce the generation of these four synthetic networks. We calculate the number of total internal and external edges if the networks are fully-connected. Internal edges indicate the edges inside the communities and external edges indicate the edges between any two of the communities. The total internal edges for such networks (100 nodes in each of 3 communities) should be $\frac{(100\times 99\times 3)}{2}$ = 14850 and the total external edges should be $\frac{(100\times 200\times 3)}{2}$ = 30000. We generate four different networks by adding different percentage of internal positive edges to make their communities range from spare-connected to densely-connected structure as shown in Table 2. We also add 10% of internal positive edges as external positive edges to be noise in these networks.

The first network is a network with sparse-connected community structure. There are only 100 internal positive edges in each community, which take 2% of total internal edges. We also add 30 external positive edges as noise, which are 0.1% of total external edges.

The second network is added 500 internal positive edges in each community, which are 10% of total internal edges. We also add 150 external positive edges, which are 0.5% of total external edges.

The third network has more densely-connected community structure. We add 2000 internal positive edges in each community, which are 40% of total internal edges. Also, 600 external positive edges are added, which are 2% of total external edges.

At last, we make an extreme case. We add 4500 internal positive edges in each community, which are 91% of total internal edges. The external positive edges are 1350, which are 4.5% of total external edges.

Table 2: The number and percentage of internal and external edges for four networks.

Networks	Internal edges / PCT	External edges / PCT
Fully-Connected	14850/100%	30000/100%
1	300/2%	30/0.1%
2	1500/10%	150/0.5%
3	6000/40%	600/2%
4	13500/90.0%	1350/4.5%

For these four networks, we firstly add external negative edges, and then we switch to add internal negative edges to study the spectral features. For both negative edges, we add them into the networks from a small percentage to a full percentage gradually. We can find that the networks change from one type of signed graph to another type with the increasing amount of negative edges.

4.2.2.2 Experiment Results

The results are shown from Figure 13 to Figure 20. We always show the spectral patterns in the first 3 dimensions in the left column of our results. The middle column shows the curves of the eigenvalues for the corresponding networks. The dimensions in these curves are sorted in descending order according to the eigenvalues. The right column shows the selected 3 dimensions based on the outstanding eigenvalues on the curves. There should be no results if the curves of the eigenvalues show nothing special, but in order to summary the changes of spectral patterns in these selected dimensions from the beginning to the end, we put these results in this column as well.

In our results, we do not show the networks with all percentage of the negative edges we added. We show the results which indicate the start or the representation of the changes. For example, Figure 13 shows we add external negative edges into the 1st network. The second row shows the community starts to form the separated clusters by adding 0.9% of external negative edges . When the percentage goes to 5%, the curve of eigenvalues shows several large absolute eigenvalues in the first two and the last one dimensions. The spectral pattern in these dimensions shows quasi-orthogonal lines. When the percentage goes to 30%, the spectral pattern in the first 3 dimensions shows 3 separated communities but expand along the 3rd dimension. While the spectral pattern in the selected 3 dimensions shows communities form 3 separated quasi-orthogonal lines. The results with this percentage are representative for the spectral patterns in a large range of percentages. When the percentage goes to 90%, the spectral pattern in the first 3 dimensions shows the communities expand along the 3rd dimension as three lines. While the spectral pattern in the first two and the last dimensions shows the communities start to cluster together. Finally, when the percentage goes to maximum 99.9%, the spectral pattern in the selected 3 dimensions shows the network structure with 3 highly-clustered communities.

Additionally, we select several cases to add more external positive edges as noises into the networks. From Figure 21, the spectral layouts in the dimensions with the largest absolute eigenvalues can explore the community structures successfully. we also generate four synthetic networks with four communities to explore the spectral pattern changes by adding external negative or internal negative edges. The results are shown from Figure 22 to Figure 25.

PCT | The spectral patterns in the (1,2,3) dimensions | The curves of Eigenvalues | The spectral patterns in the (1,2,the last) dimensions

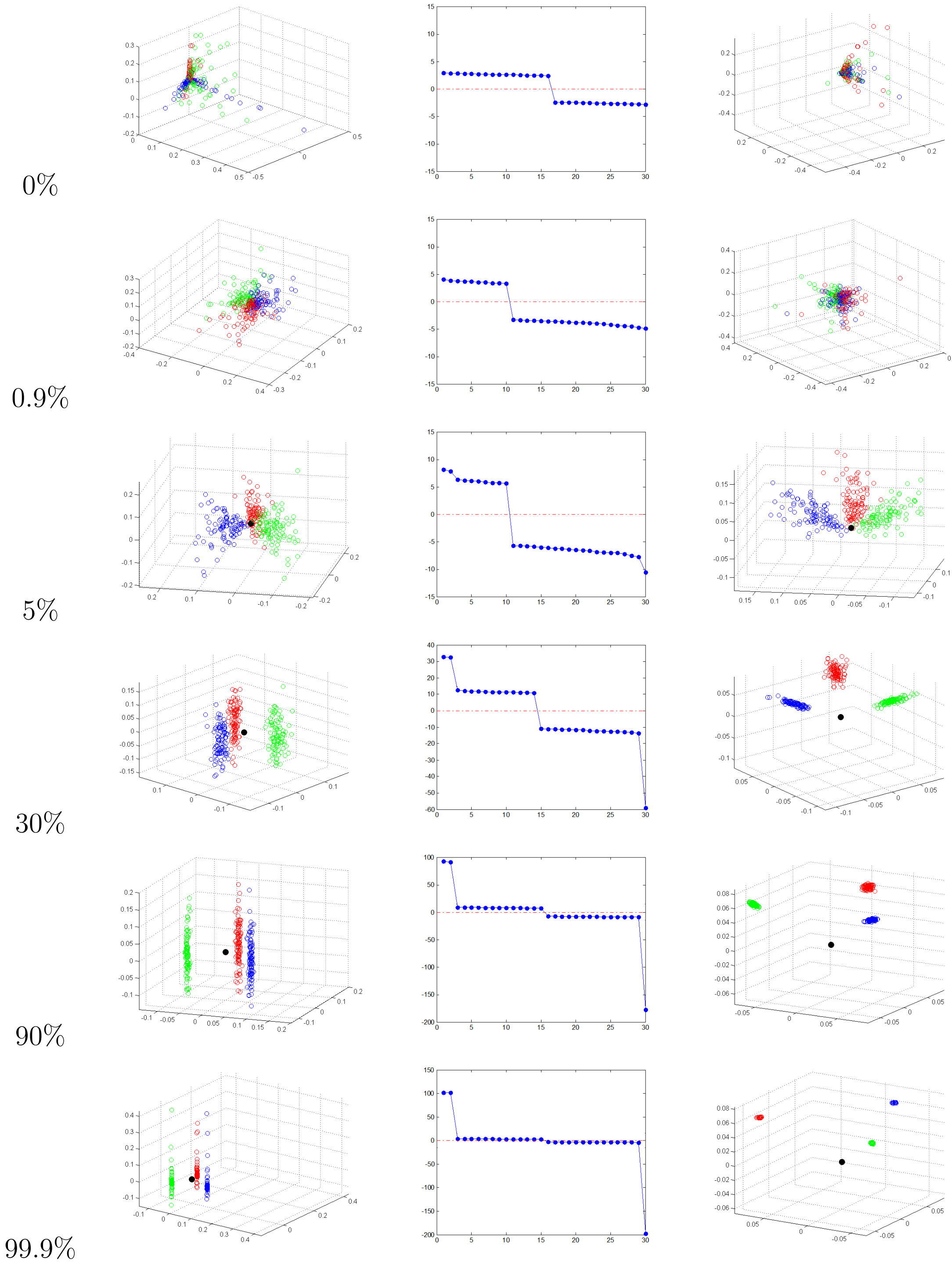


Figure 13: Spectral patterns for the 1st signed network (external -). The network has 300 internal positive edges (2% of total 14850 internal edges) and 30 external positive edges as noise(0.1% of total 30000 external edges).

PCT The spectral patterns in the (1,2,3) dimensions The curves of Eigenvalues The spectral patterns in the (1,2,the last) dimensions

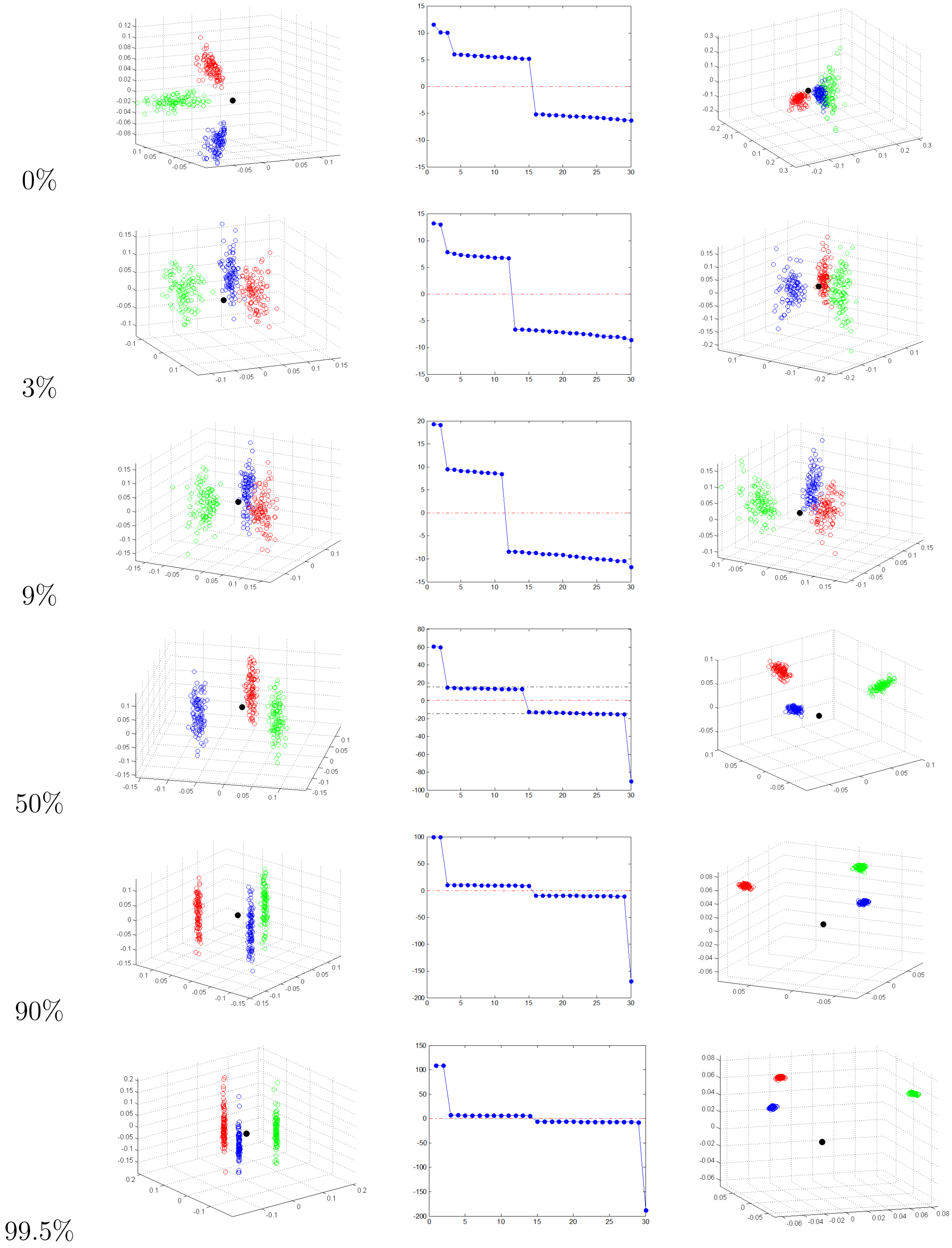


Figure 14: Spectral patterns for the 2nd signed network (external -). The network has 1500 internal positive edges (10% of total 14850 internal edges) and 150 external positive edges as noise(0.5% of total 30000 external edges).

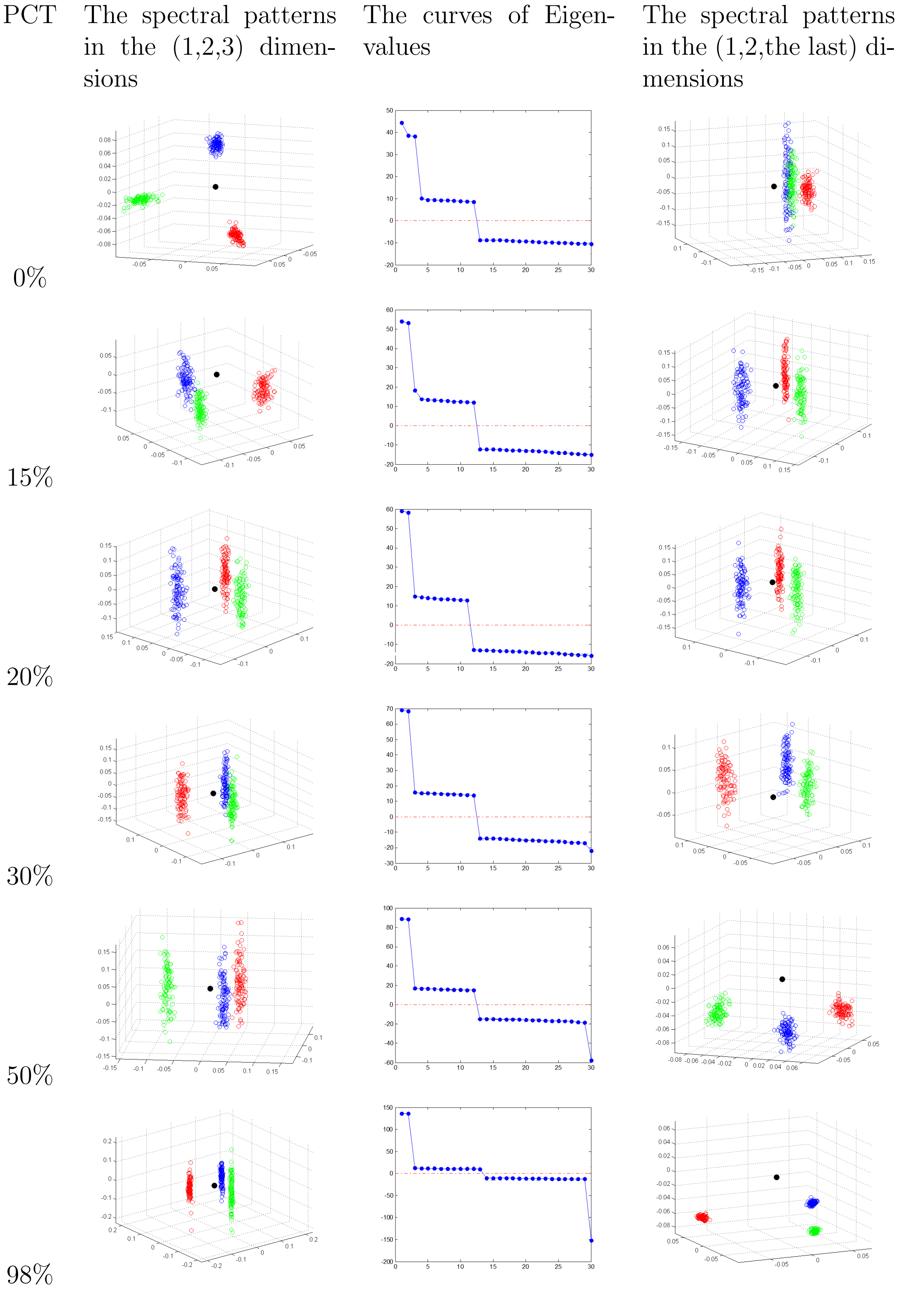


Figure 15: Spectral patterns for the 3rd signed network (external -). The network has 6000 internal positive edges (40% of total 14850 internal edges) and 600 external positive edges as noise (2% of total 30000 external edges).

PCT | The spectral patterns in the (1,2,3) dimensions | The curves of Eigenvalues | The spectral patterns in the (1,2,the last) dimensions

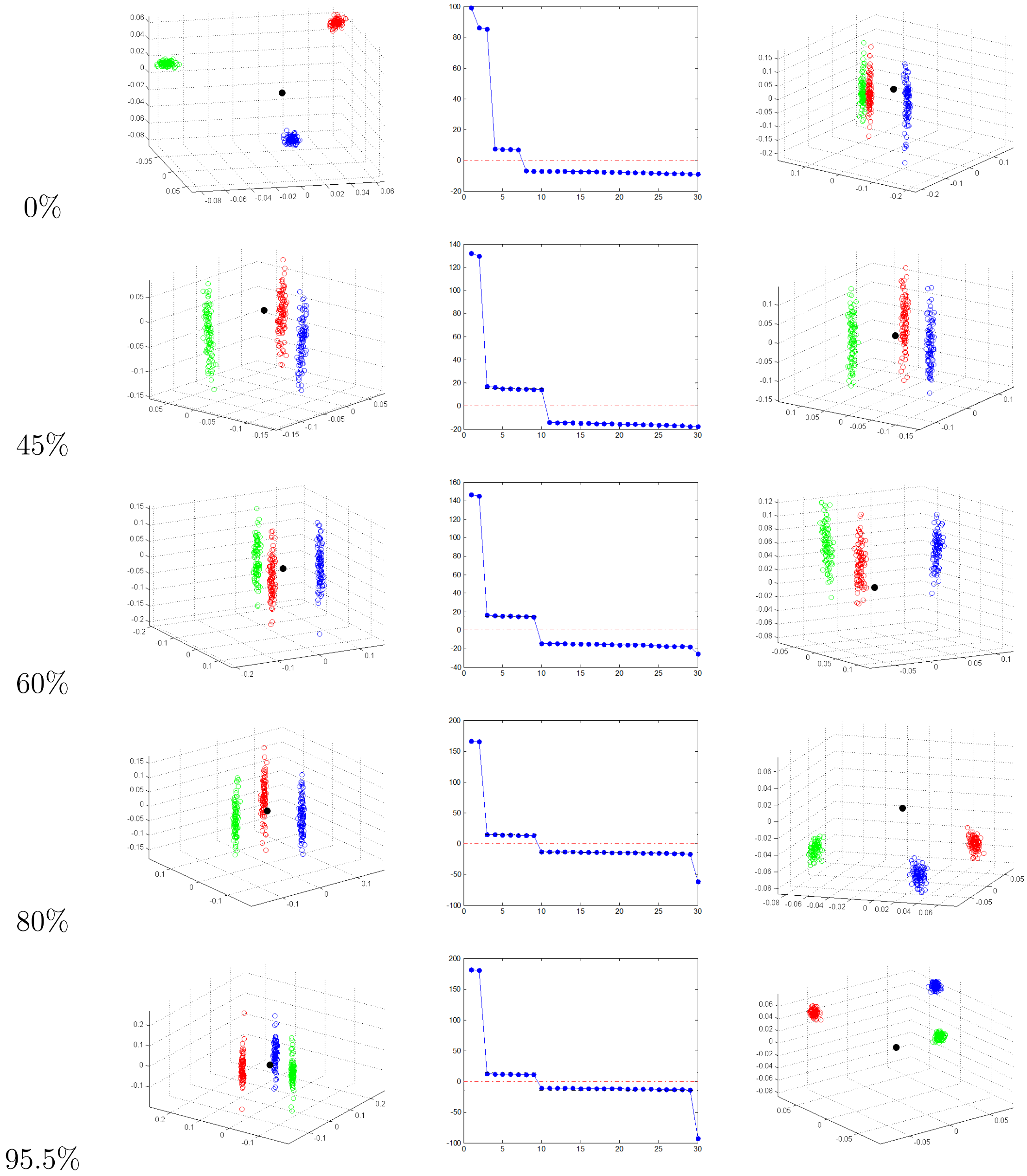


Figure 16: Spectral patterns for the 4th signed network (external -). The network has 13500 internal positive edges (90.9% of total 14850 internal edges) and 1350 external positive edges as noise (4.5% of total 30000 external edges).

PCT | The spectral patterns in the (1,2,3) dimensions | The curves of Eigenvalues | The spectral patterns in the last 3 dimensions

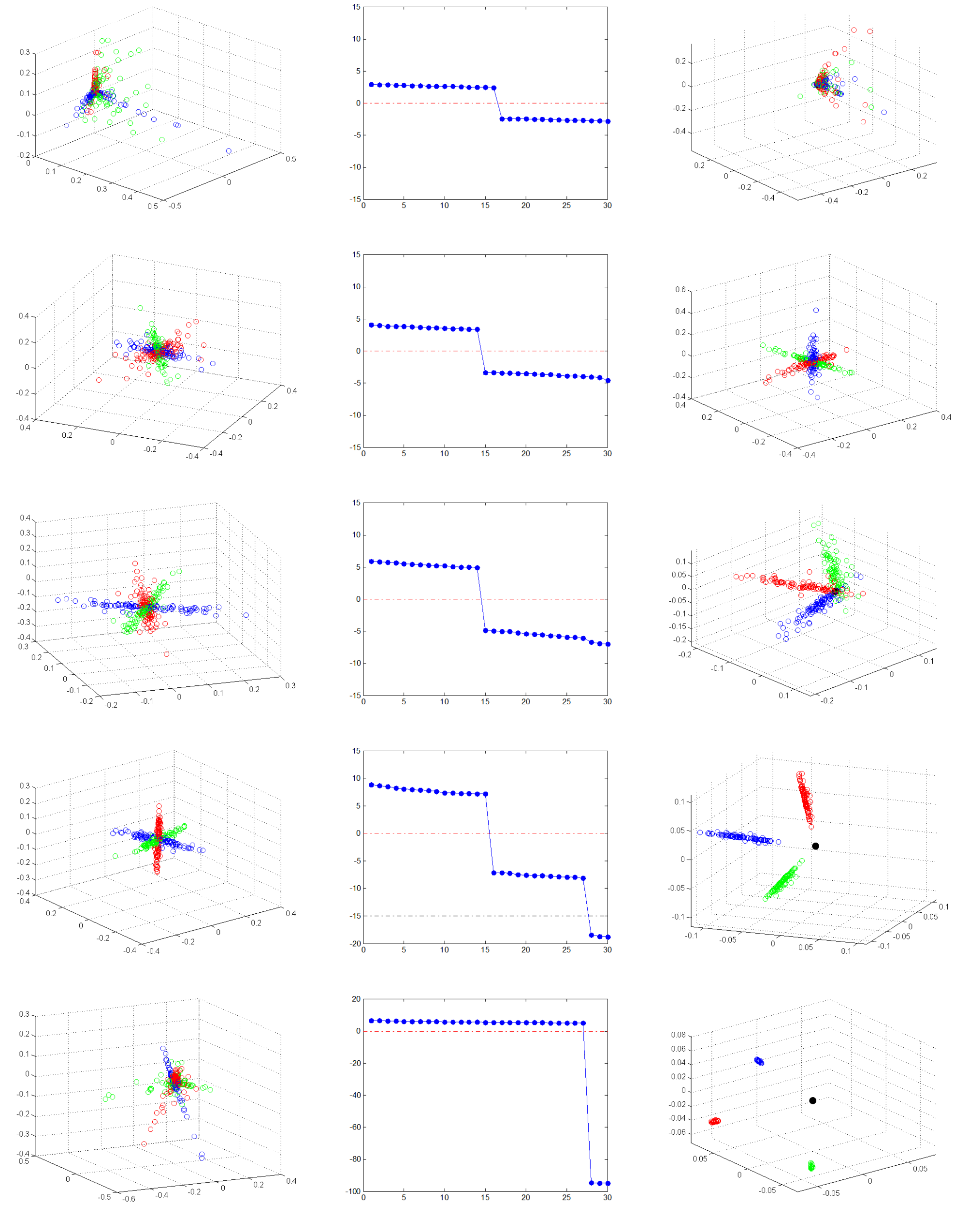


Figure 17: Spectral patterns for the 1st signed network (internal -). The network has 300 internal positive edges (2% of total 14850 internal edges) and 30 external positive edges as noise(0.1% of total 30000 external edges).

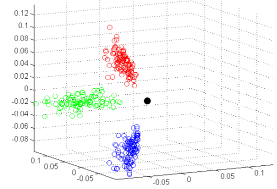
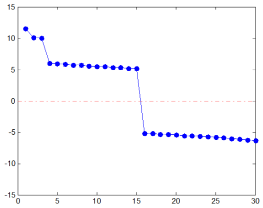
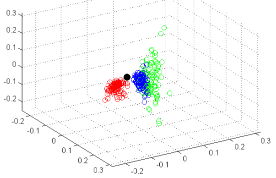
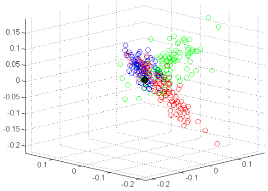
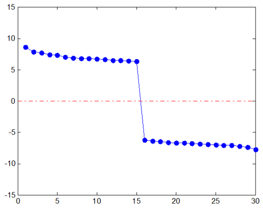
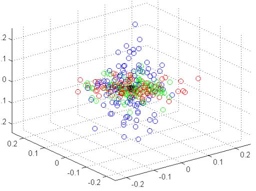
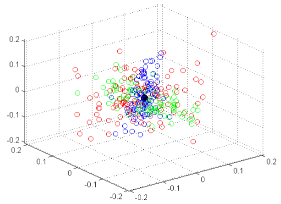
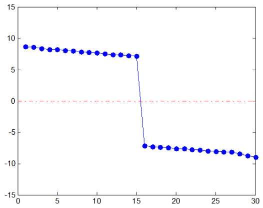
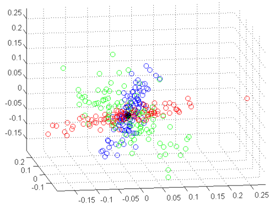
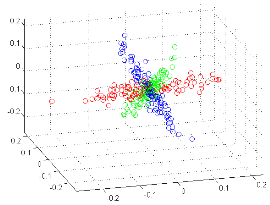
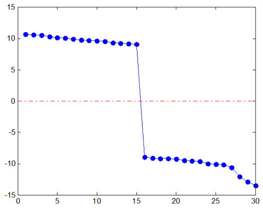
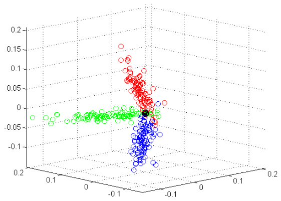
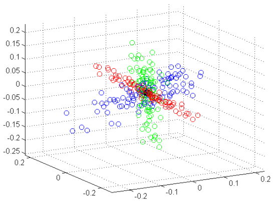
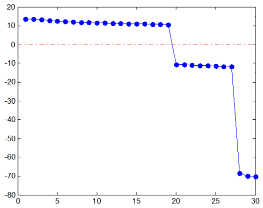
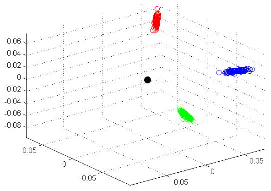
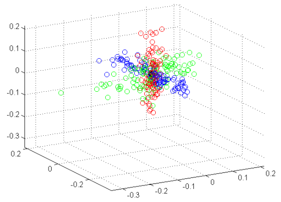
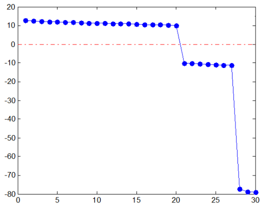
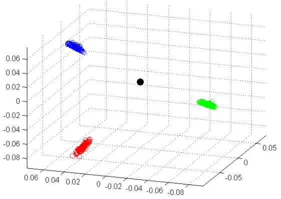
PCT	The spectral patterns in the (1,2,3) dimensions	The curves of Eigenvalues	The spectral patterns in the last 3 dimensions
0%			
4%			
9%			
20%			
80%			
90%			

Figure 18: Spectral patterns for the 2nd signed network (internal -). The network has 1500 internal positive edges (10% of total 14850 internal edges) and 150 external positive edges as noise(0.5% of total 30000 external edges).

PCT | The spectral patterns in the (1,2,3) dimensions | The curves of Eigenvalues | The spectral patterns in the last 3 dimensions

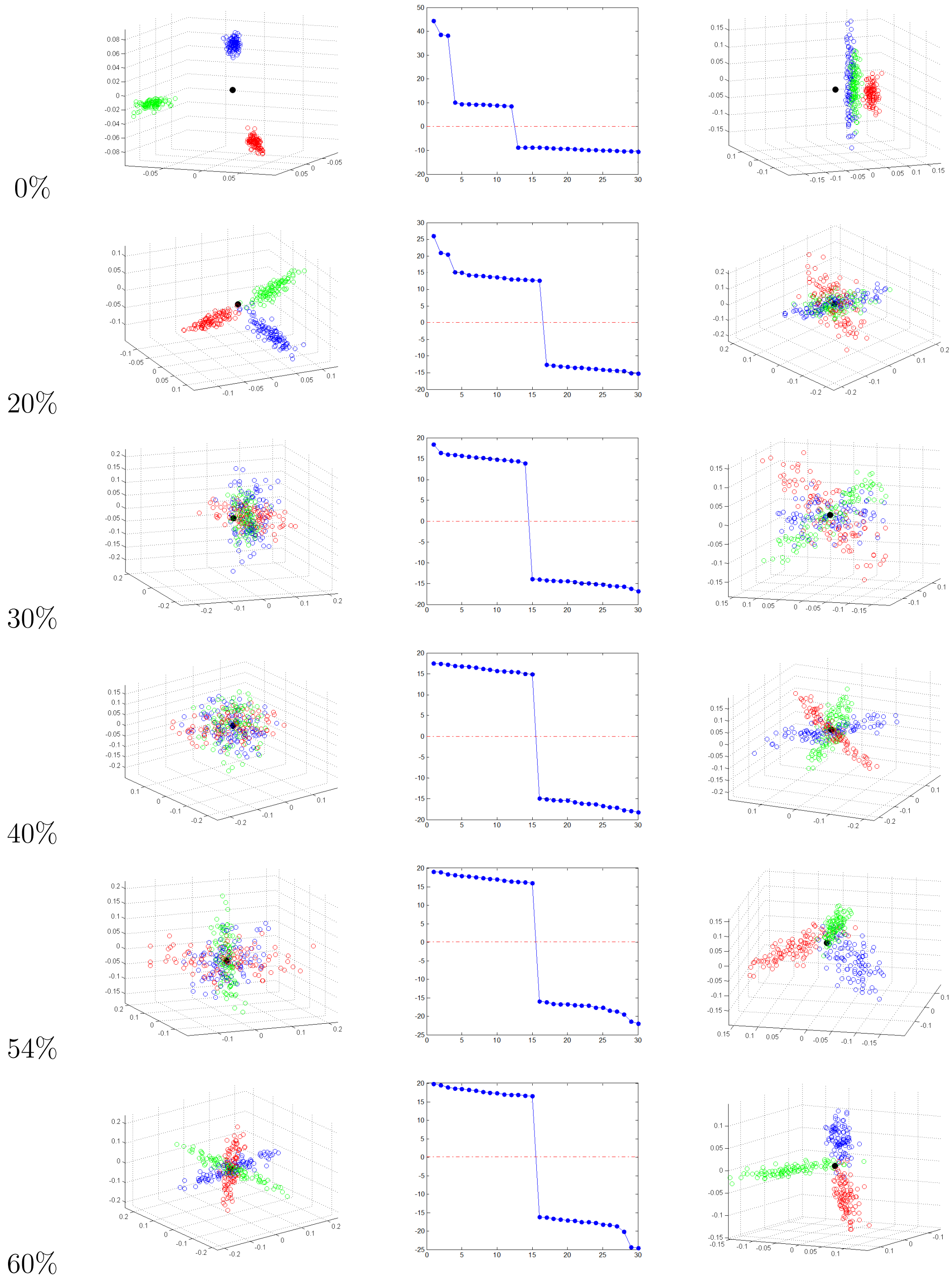


Figure 19: Spectral patterns for the 3rd signed network (internal -). The network has 6000 internal positive edges (40% of total 14850 internal edges) and 600 external positive edges as noise (2% of total 30000 external edges).

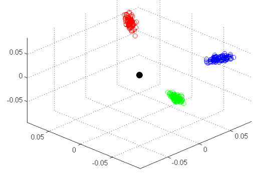
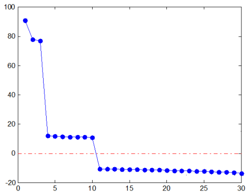
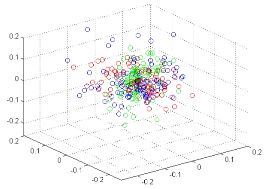
PCT	The spectral patterns in the (1,2,3) dimensions	The curves of Eigenvalues	The spectral patterns in the last 3 dimensions
9%			

Figure 20: Spectral patterns for the 4th signed network (internal -). The network has 13500 internal positive edges (90.9% of total 14850 internal edges) and 1350 external positive edges as noise (4.5% of total 30000 external edges).

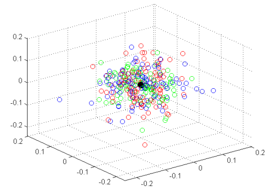
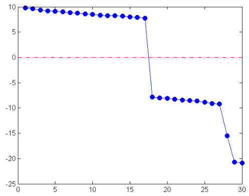
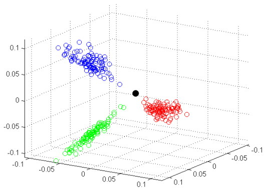
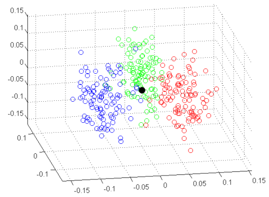
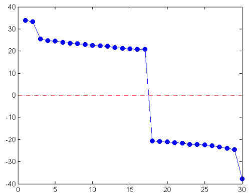
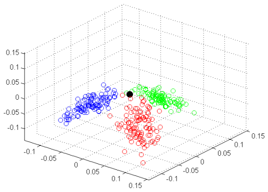
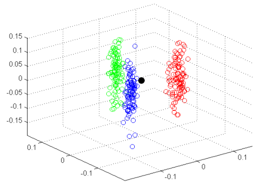
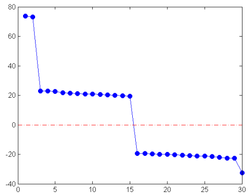
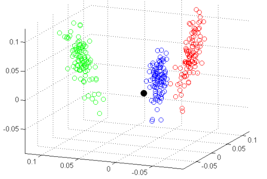
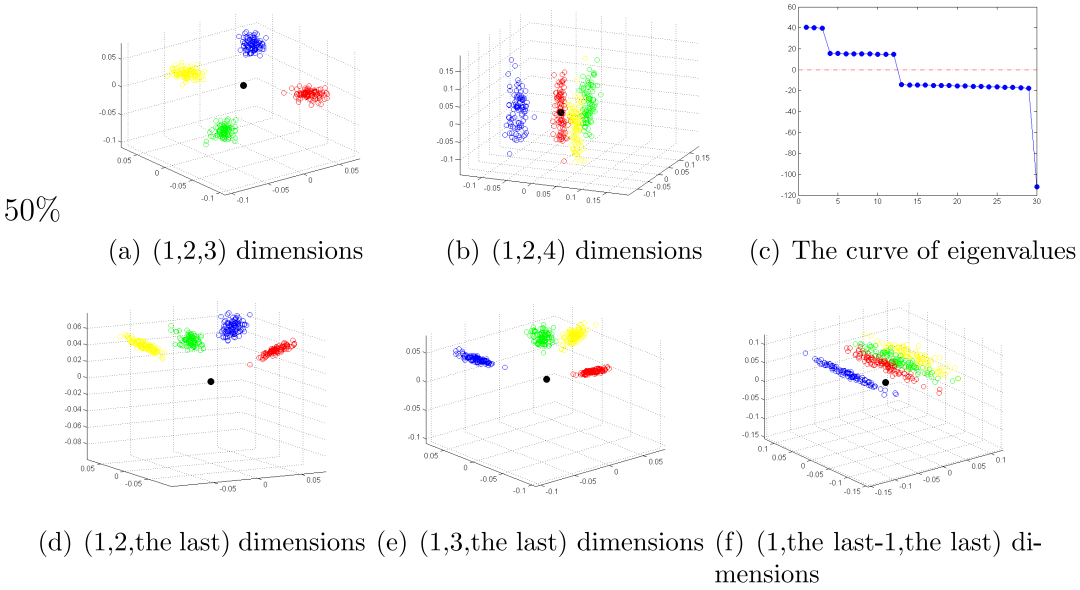
PCT	The spectral patterns in the (1,2,3) dimensions	The curves of Eigenvalues	The spectral patterns in 3 dimensions with the largest absolute eigenvalues
20%			
50%			
50%			

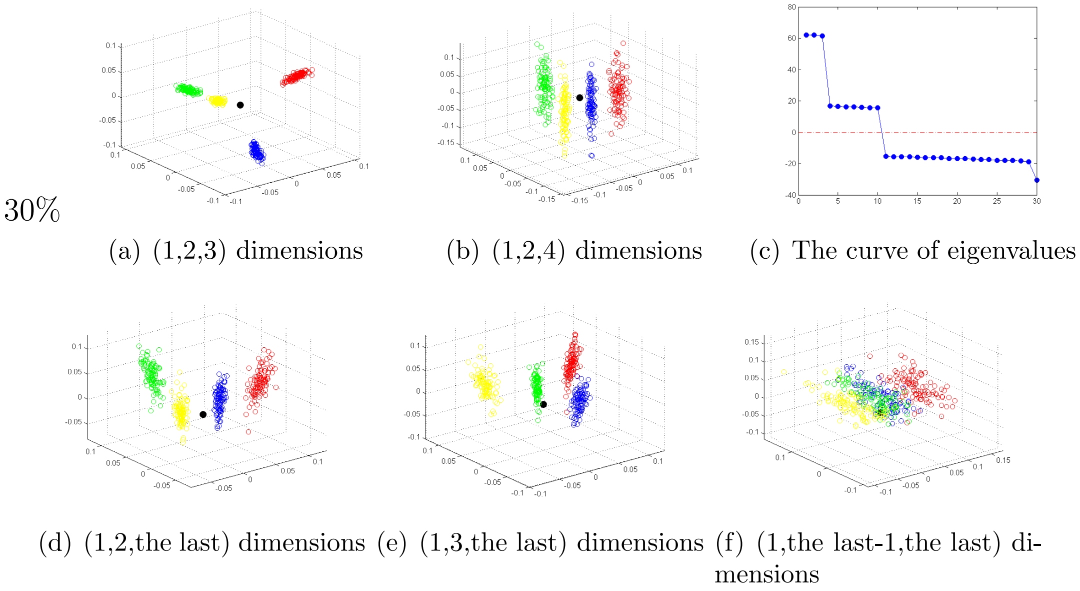
Figure 21: Spectral patterns for the signed networks with more noise edges. The upper network has 300 internal positive edges (2% of total 14850 internal edges) and 600 external positive edges as noise (2% of total 30000 external edges). We also add 20% of internal negative edges. The middle network has 1500 internal positive edges (10% of total 14850 internal edges) and 9000 external positive edges as noise (30% of total 30000 external edges). We also add 50% of external negative edges. The bottom network has 6000 internal positive edges (40% of total 14850 internal edges) and 5400 external positive edges as noise (18% of total 30000 external edges). We also add 50% of external negative edges.



(a) (1,2,3) dimensions (b) (1,2,4) dimensions (c) The curve of eigenvalues

(d) (1,2,the last) dimensions (e) (1,3,the last) dimensions (f) (1,the last-1,the last) dimensions

Figure 22: Spectral patterns for the 1st signed networks with four communities. The upper network has 400 internal positive edges (2% of total 19800 internal edges) and 40 external positive edges as noise (0.09% of total 45000 external edges). We also add 50% of external negative edges.

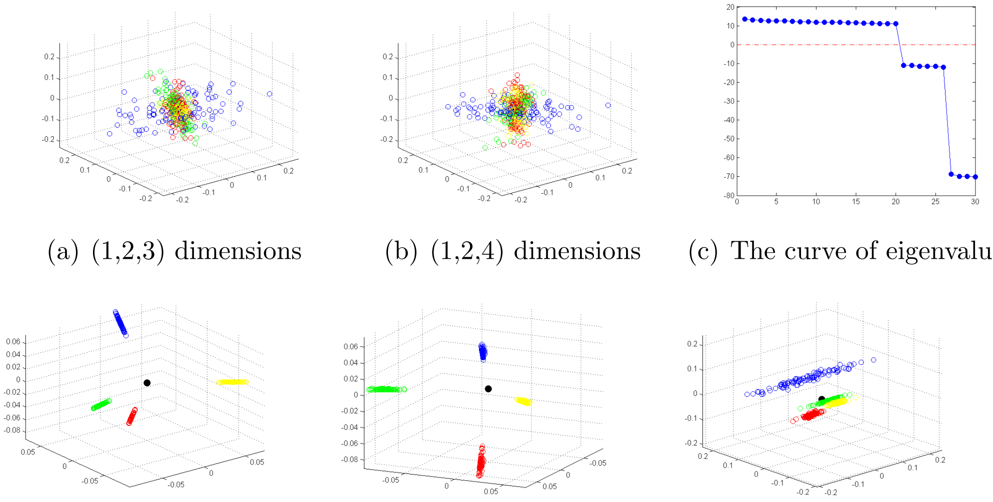


(a) (1,2,3) dimensions (b) (1,2,4) dimensions (c) The curve of eigenvalues

(d) (1,2,the last) dimensions (e) (1,3,the last) dimensions (f) (1,the last-1,the last) dimensions

Figure 23: Spectral patterns for the 2nd signed networks with four communities. The upper network has 8000 internal positive edges (41% of total 19800 internal edges) and 800 external positive edges as noise (1.8% of total 45000 external edges). We also add 30% of external negative edges.

80%

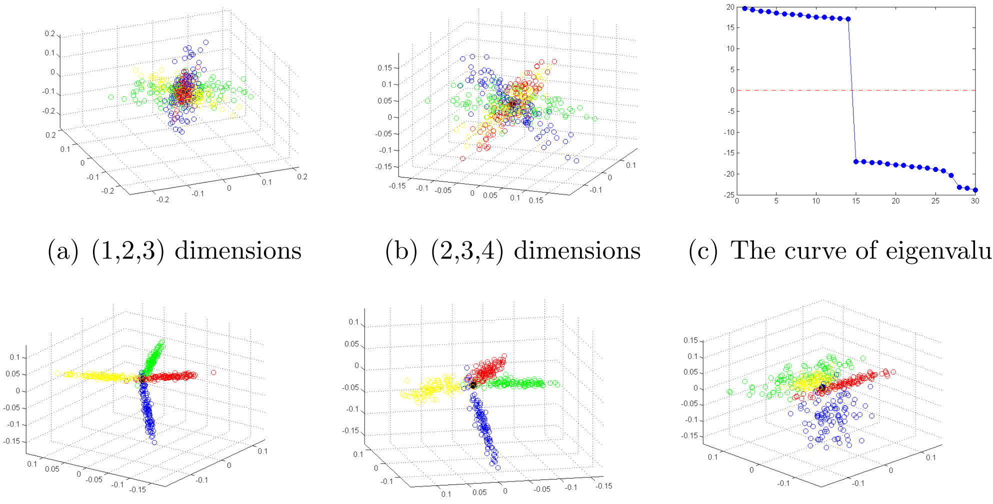


(a) (1,2,3) dimensions (b) (1,2,4) dimensions (c) The curve of eigenvalues

(d) (the last-2,the last-1,the last) dimensions (e) (the last-3,the last-2,the last) dimensions (f) (1,the last-1,the last) dimensions

Figure 24: Spectral patterns for the 3rd signed networks with four communities. The bottom network has 2000 internal positive edges (10% of total 19800 internal edges) and 200 external positive edges as noise (0.4% of total 45000 external edges). We also add 80% of internal negative edges.

59%



(a) (1,2,3) dimensions (b) (2,3,4) dimensions (c) The curve of eigenvalues

(d) (the last-2,the last-1,the last) dimensions (e) (the last-3,the last-2,the last) dimensions (f) (1,the last-1,the last) dimensions

Figure 25: Spectral patterns for the 4th signed networks with four communities. The bottom network has 8000 internal positive edges (41% of total 19800 internal edges) and 800 external positive edges as noise (1.8% of total 45000 external edges). We also add 59% of internal negative edges.

4.2.2.3 Summary

In conclusion, by adding external negative edges, the spectral patterns change from quasi-orthogonal lines or blocks to parallel lines along the third or fourth dimension. The eigenvalue curves always contain several outstanding absolute values, but they change from several positive values to several positive and 1 negative value. By adding internal negative edges, the spectral patterns change from quasi-orthogonal lines or blocks to several quasi-orthogonal lines crossing at the origin of the spectral space. Several outstanding eigenvalues change from all positive to all negative.

4.2.3 k-block Network

Definition: A k-block signed network represents a graph with k communities such that 1) inside each community, nodes are densely connected with the same signs; and 2) there are no links between different communities. The adjacency matrix A_b of a k-block signed network can be written in the following form with proper permutation of the nodes:

$$A_b = \begin{pmatrix} A_1 & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & A_k \end{pmatrix}, \tag{7}$$

where A_i is the $n_i \times n_i$ adjacency matrix of the ith community with n_i nodes. We call A_b as a k-block matrix.

Our first result proves that the quasi-orthogonality theorem holds when applied to k-block signed networks. The proof is provided below.

Result 1: The k-block signed network shows k orthogonal lines in the k-dimensional spectral subspace spanned by $\boldsymbol{x}_i$'s of the adjacency matrix with corresponding eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_k|$ when the k communities are comparable and $|\lambda_k| \gg |\lambda_{k+1}|$. Specifically, when there are k_1 blocks with non-negative entries and k_2 blocks with non-positive entries ($k_1 + k_2 = k$), the subspace is spanned by $\boldsymbol{x}_i$'s with the first k_1 largest eigenvalues and the first k_2 largest absolute values.

Proof of Result 1: With proper relabeling the nodes, we can write a k-block matrix into

$$A_b = \begin{pmatrix} A_b^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_b^- \end{pmatrix} \tag{8}$$

where A_b^+ contains all k_1 blocks with non-negative entries and A_b^- contains all k_2 blocks with non-positive entries.

Let λ_i^+ and $\boldsymbol{x}_i^+$ be the ith eigenvalue and eigenvector of A_b^+ where $\lambda_1^+ > \lambda_2^+ > \cdots > \lambda_n^+$. By Lemma 2 in [134], if the k_1 communities are comparable and $\lambda_{k_1}^+ \gg |\lambda_{k_1+1}^+|$, the corresponding k_1 eigenvectors of A_b^+ have the following form:

$$(\boldsymbol{x}_1^+, \boldsymbol{x}_2^+, \cdots, \boldsymbol{x}_k^+) = \begin{pmatrix} \boldsymbol{x}_{A_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{x}_{A_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{x}_{A_{k_1}} \end{pmatrix}, \tag{9}$$

Similarly, we can apply the Lemma on $-A_b^-$. Let λ_i^- and $\boldsymbol{x}_i^-$ be the ith eigenvalue and eigenvector of A_b^- where $\lambda_1^- < \lambda_2^- < \cdots < \lambda_n^-$. Since $(-A_b^-)\boldsymbol{x}_i^- = -(A_b^-\boldsymbol{x}_i^-) = -(\lambda_i^-\boldsymbol{x}_i^-) = (-\lambda_i^-)\boldsymbol{x}_i^-$, $-\lambda_1^-, \cdots, -\lambda_{k_2}^-$ are the largest k_2 eigenvalues for $-A_b^-$. Again when the k_2 communities are comparable and $-\lambda_{k_2}^- \gg |\lambda_{k_2+1}^-|$, k_2 eigenvectors of A_b^-

have the following form:

$$(\boldsymbol{x}_1^-, \boldsymbol{x}_2^-, \cdots, \boldsymbol{x}_k^-) = \begin{pmatrix} \boldsymbol{x}_{A_{k_1+1}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{x}_{A_{k_1+2}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{x}_{A_k} \end{pmatrix},$$

Due to the block structure, the eigenvalues of blocks are the eigenvalues of the whole matrix and the eigenvectors of the blocks are nonzero part of the eigenvectors.

$$\begin{aligned} A_b \begin{pmatrix} \boldsymbol{x}_i^+ \\ \mathbf{0} \end{pmatrix} &= \begin{pmatrix} A_b^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_i^+ \\ \mathbf{0} \end{pmatrix} + \begin{pmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_b^- \end{pmatrix} \begin{pmatrix} \boldsymbol{x}_i^+ \\ \mathbf{0} \end{pmatrix} \\ &= \begin{pmatrix} A_b^+ \boldsymbol{x}_i^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \begin{pmatrix} \lambda_i^+ \boldsymbol{x}_i^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{pmatrix} = \lambda_i^+ \begin{pmatrix} \boldsymbol{x}_i^+ \\ \mathbf{0} \end{pmatrix} \end{aligned}$$

Similarly,

$$A_b \begin{pmatrix} \mathbf{0} \\ \boldsymbol{x}_i^- \end{pmatrix} = \lambda_i^- \begin{pmatrix} \mathbf{0} \\ \boldsymbol{x}_i^- \end{pmatrix}$$

So $\lambda_1^+, \cdots, \lambda_{k_1}^+$ and $\lambda_{k1}^-, \cdots, \lambda_{k_2}^+$ are the k eigenvalues of A_b that have the corresponding eigenvectors with the form:

$$(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_k) = \begin{pmatrix} \boldsymbol{x}_{A_1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{x}_{A_2} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{x}_{A_k} \end{pmatrix}, \tag{10}$$

Since the communities are comparable, we have $\min\{|\lambda^+_{k_1}|, |\lambda^-_{k_2}|\} \gg \max\{|\lambda^+_{k_1+1}|, |\lambda^-_{k_2+1}|\}$, so that the k eigenvalues are the k ones with largest absolute value.

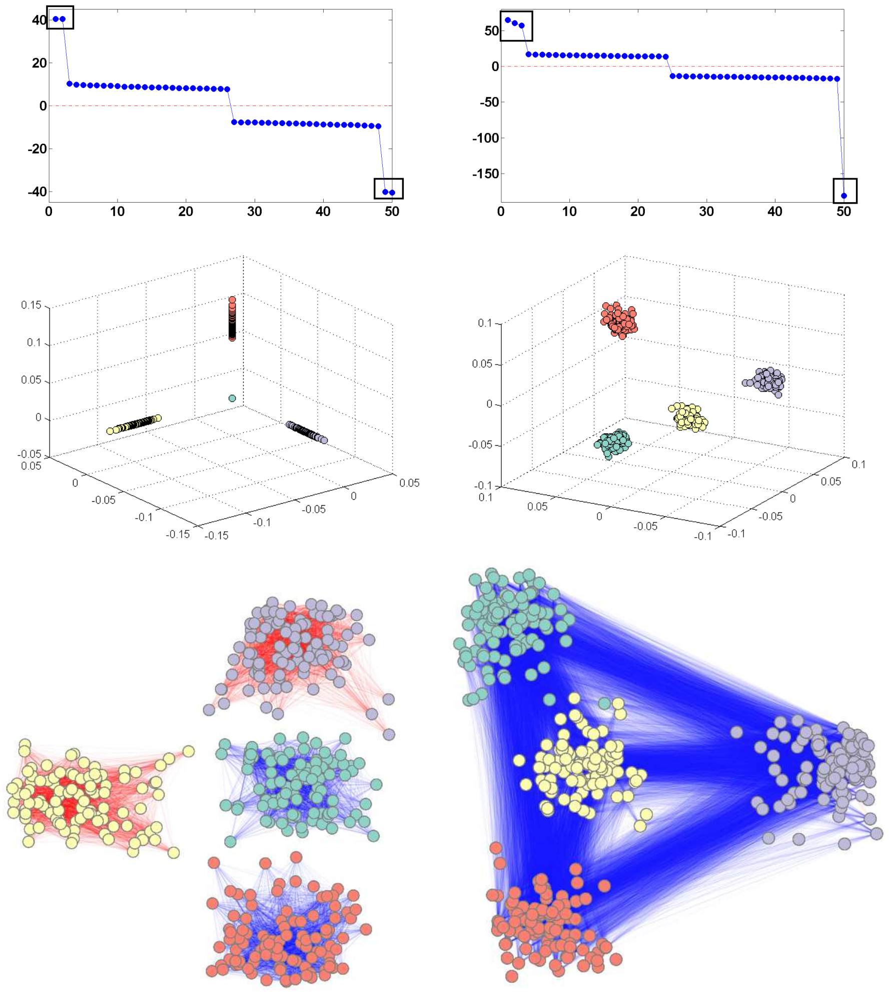
For node u, $\alpha_u = (\mathbf{0}', x_{ui}, \mathbf{0}')$. It stays on one of the axis in the k dimensional subspace spanned by $\boldsymbol{x}_i$'s. So the nodes form k orthogonal lines.

Patterns: For a k-block signed network with k_1 blocks non-negative and k_2 blocks non-positive, we can find k eigenvalues with large absolute values. The number of large positive eigenvalues (k_1) indicates the number of communities with dense positive internal relationships and the number of negative eigenvalues (k_2) indicates the number of communities with dense negative internal relationships. Specifically, a k-block network with all non-negative entries has k large positive eigenvalues and node coordinates form k orthogonal lines in the subspace spanned by their corresponding eigenvectors. In contrast, for a k-block network with all non-positive entries, we have similar conclusion: node coordinates form k orthogonal lines in the subspace spanned by eigenvectors corresponding to k large negative eigenvalues.

The left column of Figure 26 provides an example of the k-block network. This network contains four communities with 100 nodes each. Two communities have 2000 positive edges and the other two have 2000 negative edges each. The curve of eigenvalues reveals two positive outstanding eigenvalues and two negative outstanding eigenvalues, which is consistent with our spectral analysis results above.

4.2.4 k-partite Network

The k-partite network describes the relationships of nodes between different communities. We first provide the definition and then the study result.



k-block network k-partite network

Figure 26: Examples of the two signature signed networks. Eigenvalue curves 50 dimensions, spectral patterns in the selected subspaces, and our network layout results are shown. Positive edges are in red and negative edges are in blue.

Definition: A k-partite network represents a graph with k communities such that 1) there are no links inside the communities; and 2) nodes from different communities are densely connected with the same signs. The adjacency matrix A_p can be written in the following form with proper permutation of the nodes:

$$A_p = \begin{pmatrix} \mathbf{0} & B_{12} & \cdots & B_{1k} \\ B_{21} & \mathbf{0} & \cdots & B_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ B_{k1} & B_{k2} & \cdots & \mathbf{0} \end{pmatrix}, \tag{11}$$

where B_{ij} is the $n_i \times n_j$ matrix to represent the relationships between community i and community j. We call A_p as a k-partite matrix.

In [131], the authors showed the approximation forms of eigenvectors and spectral coordinates for k-partite matrix. They proved that such a matrix shows k orthogonal clusters when the communities have similar densities and the first eigenvalue has a different sign with the following $k-1$ eigenvalues in magnitude.

Specially, the k-partite network with k comparable communities shows k orthogonal clusters in the k-dimensional spectral subspace spanned by $\boldsymbol{x}_i$'s of the adjacency matrix with corresponding eigenvalues $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_k|$. Furthermore, $\|\lambda_1\|$ has a different sign with the rest $k-1$ eigenvalues.

The right column of Figure 26 provides an example of the k-partite network. This network contains four communities with 400 nodes and 100 positive internal edges inside each communities. There are 36000 negative external edges added randomly between communities. As shown in the curve of eigenvalues, there are three very high positive eigenvalues and one very low negative eigenvalue, representing the four communities in the network.

4.2.5 General Signed Networks

Our work concentrates on studying spectral features of signed networks related to the number of communities, represented as k in this chapter, and structures of communities. Here, communities are loosely defined as collections of network nodes that interact unusually frequently, including both positive and negative relationships. In practice, general signed networks may be involved of complex internal and external relationships. The adjacency matrix A can be written in the following form with proper permutation of the nodes:

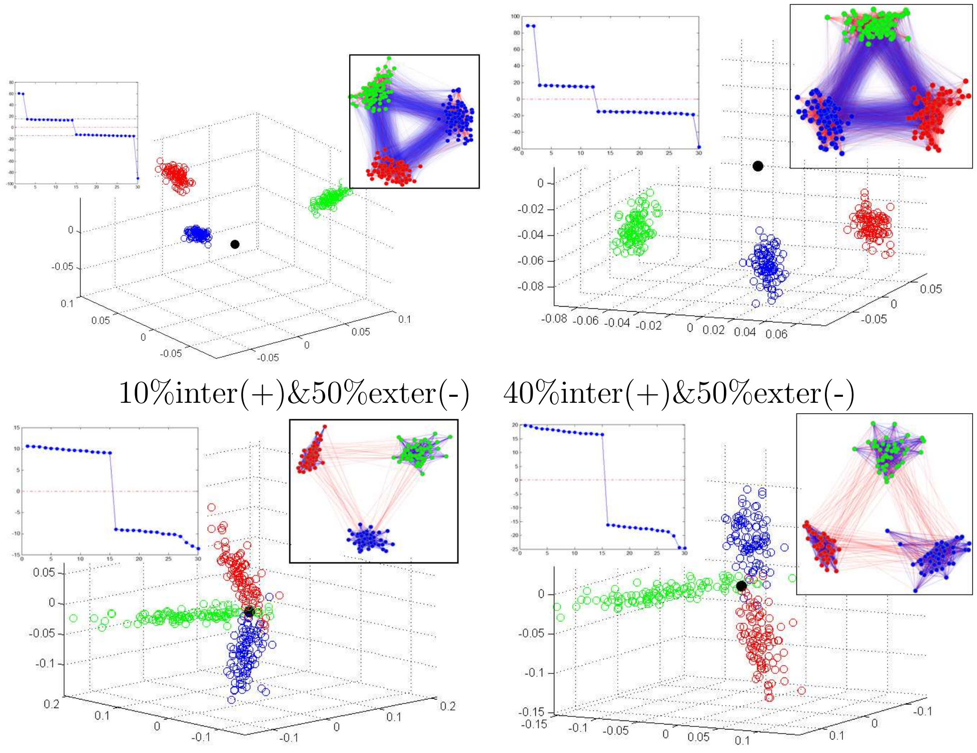
$$A = \left(\begin{array}{cccccc} \boldsymbol{A_1} & B_{12} & \cdots & | & \cdots & B_{1k} \\ B_{21} & \boldsymbol{A_2} & \cdots & | & \cdots & B_{2k} \\ \vdots & \vdots & \cdots & | & \cdots & \vdots \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ \vdots & \vdots & \cdots & | & \cdots & \vdots \\ B_{k1} & B_{k2} & \cdots & | & \cdots & \boldsymbol{A_k} \end{array} \right), \tag{12}$$

where the definitions of A_i and B_{ij} are the same as in Formulas 10 and 11. We argue that our spectral analysis results of the two signature signed networks can still be used to study general signed networks from the following three aspects.

First, spectral analysis always presents the dominant community structures in the networks. Eigen-decomposition produces an indexed set of linearly independent eigenvectors, where the first eigenvector having the direction of largest variance of the data. No matter how complex a network is, the dominant community structures are always revealed on the first several dimensions. This is consistent with that the community relationships of complex networks can be represented as a hierarchical structure. Therefore, the results of spectral analysis can be applied to both simple and complex networks.

Second, we discuss general signed networks where positive or negative edges dominant. While the global community structure may be complex, it can be decomposed to local communities with structures similar to one of the two signature networks. For example, as shown in Formula 12, the first group of communities may appear as a k-partite network and the second group appears as k-block networks. Also, the edge densities between these local structures should be much smaller than the densities inside each local community. Otherwise, two communities with both strong internal and external connections, no matter their signs, should be treated as one community instead. Therefore, even though a general signed network may contain a complex hierarchical community structure, it can be decomposed to a number of k-block and k-partite networks.

Third, we discuss general signed networks with various combinations of positive and negative edges. As shown in our experiment, the patterns of node distributions in the spectral space adjust gradually when the ratios of negative to positive edges change. Even for the cases whose positive and negative edges are comparable, especially when both positive and negative edges are large enough, the spectral features of both signed and unsigned networks can be shown.



10%inter(+)&50%exter(-)   40%inter(+)&50%exter(-)

10%inter(+)&20%inter(-)   40%inter(+)&60%inter(-)

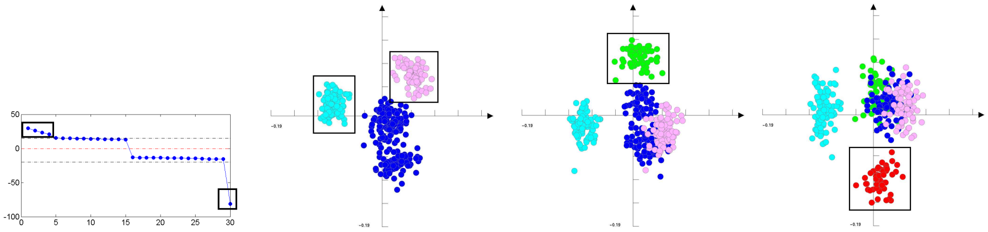
Figure 27: Spectral patterns of general signed networks can be revealed by selecting a suitable k subspace.

4.3 Spectrum-based Visualization for Signed Networks

To study general signed networks, visualization is important to explore various visual patterns in the spectral space. The main problem of visualizing a k-dimensional spectral space is that it cannot be visualized or interacted directly. This requires users to visualize ${C_k}^2$ 2-dimensional or ${C_k}^3$ 3-dimensional subspaces, which do not provide a good overview of the original k-dimensional space. Small multiple displays [94]allow users to visualize several 2D or 3D subspaces simultaneously, but users need to be aware of the dimensions used in multiple subspaces. Especially for complex networks, where k is large and not clear, it is inefficient to rely on visualization of individual subspaces of the k-dimensional spectral space.

Our motivation is to design a network layout algorithm which also provides an interaction domain of the spectral space. The network layout algorithm concentrates on revealing important community structures in the k-dimensional spectral space. The interaction methods allow users to explore effects of different combinations of the k selected eigenvectors and discover sub-communities.

Our design principal is to provide an overview of the network community structure by combining important features from multiple subspaces of the k-dimensional spectral space. The idea comes from an interactive exploration process that we used previously as shown in Figure 28. This synthetic dataset contains five communities with 50, 60, 70, 80 and 90 nodes respectively. There are 373 positive internal edges, 71 negative internal edges, 153 positive external edges, and 14486 negative external edges. The curve of eigenvalues of this network shows the k equals to five with outstanding dimensions that we can selected. We start with the 2D projection of 1-2 dimensions (Figure 28(b)) and mark two communities, shown with the bounding boxes. We proceed to 1-3 dimensions (Figure 28(c)) and 1-4 dimensions (Figure 28(d)) and mark two additional communities. The approach is to identify a minimum set of subspaces which contain important spectral features and synthesize a new interaction space for visual exploration, which are described in the following respectively.



(a) The curve (b) (1,2) (c) (1,3) (d) (1,4)

Figure 28: Community structure in different dimensions.

4.3.1 Selection of Representative Sub-Spaces

The first step is to estimate the community number k and identify a k-dimensional spectral space for visualization and interactive exploration. These two parameters are essential to analyze community structures for both visualization and analysis algorithms. They provide the starting point for visual exploration, which guides users to study visual patterns in the k-dimensional spectral space. We generally choose the number of eigenvalues with the maximum absolute values as k and their corresponding eigenvectors as the k-dimensional spectral space. For example, Figure 26 contains a number of successful examples.

The second step is to assess subspaces from the selected k-dimensional spectral space. As community structures of general networks can be represented as hierarchical trees, we choose a robust algorithm, hierarchical clustering [95], to estimate the possibility of clear community structures. We use 2D subspaces since 2D planes are intuitive and convenient for users to visualize. Any other sizes of subspaces are also applicable to our approach. The candidates of the subspaces are ${C_k}^2$ 2D planes S_{ij} composed by eigenvectors x_i and x_j. The hierarchical clustering with level k is performed on every subspace candidate. The results of the algorithm provide the cophenetic correlation coefficient to evaluate cluster fitness (as $cf(S_{ij})$) and the group information for each subspace.

The third step is to select representative subspaces for visual analysis. We measure the similarity of different subspaces through comparing their clustering results. We only need to count the number of node pairs which appear in the same group in one subspace but in different groups in the other subspace. The difference of two subspaces is measured as the average of the counts by switching the order of the subspaces. To accelerate the process, we sort the subspaces according to their cophenetic correlation coefficients and remove the spaces with low values. As every eigenvector appear in $k-1$ subspaces, it is safe to reduce the number of subspaces significantly. Since our objective is to identify representative subspaces to reduce the k-dimensional spectral space for visualization, the subspace pairs S_{ij} and S_{kl} with high difference values of $DS(S_{ij}, S_{kl})$ are selected.

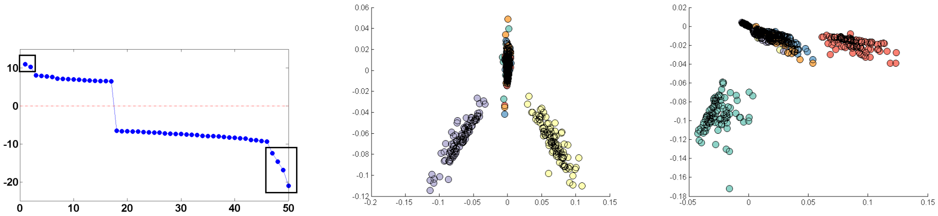
4.3.2 Generation of the Network layout

With a set of representative subspaces, we adjust the weights of individual eigenvectors on the network layout. This is achieved by selecting a single subspace for each eigenvector, indicating important community structures can be identified from this eigenvector at the subspace. We sort all the difference values of subspace pairs. Start from the largest difference value, the subspaces are assigned to the x and y dimensions respectively. The process completes until all the k eigenvectors are included.

The network layout is generated by synthesizing the representative subspaces linearly. The weights are the combination of the cophenetic correlation coefficients, the differences of subspaces, and scale values assigned by users $w(x_i)$ for interactive exploration. The new coordinate (x', y') of a node is computed for the two dimensions respectively as follows:

$$
\begin{aligned}
x' &= \sum w(x_i) \times cf(S_{ij}) \times DS(S_{ij}, S_{kl}) \times x_i \\
y' &= \sum w(x_j) \times cf(S_{ij}) \times DS(S_{ij}, S_{kl}) \times x_j
\end{aligned}
\tag{13}
$$

Figure 29 provides an example of the network layout approach. The network has 600 nodes with the structure of two positive communities and four negative communities, which can be discovered from the curve of eigenvalues in Figure 29. Since this network is mixed with k-block and k-partite structures, it is not easy to project all spectral features in such a 2D space. Actually, any 2D subspaces in the spectral space with the six outstanding eigenvalues can only partially separate communities, as examples provided on Figure 29. We can combine our interactive exploration approach to visualize all the six communities.

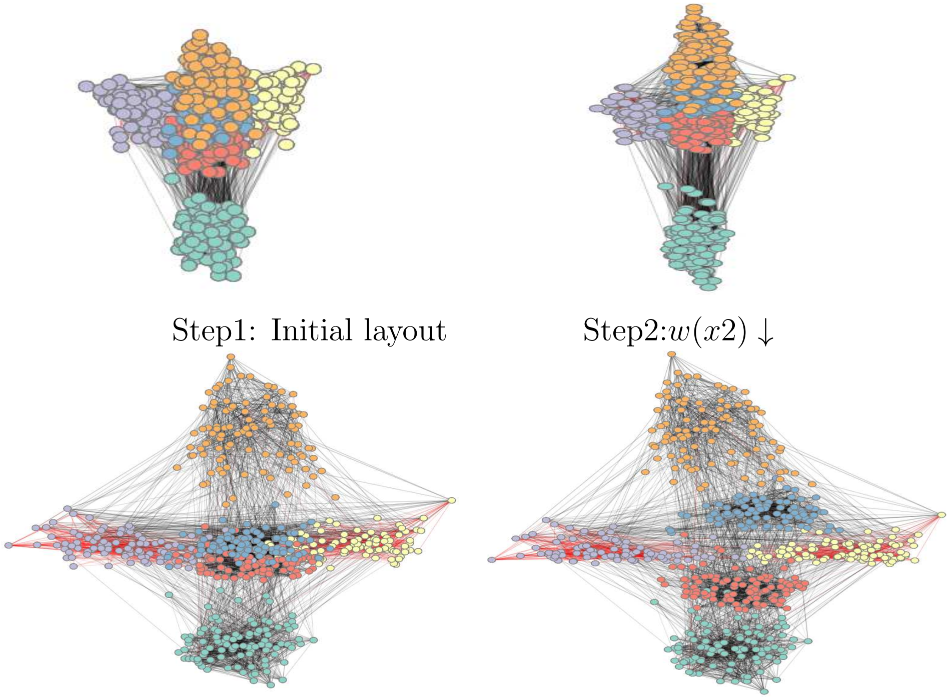


Eigenvalues (1,2) (49,50)

Figure 29: The demonstration of spectral layouts in the selected dimensions.

4.3.3 Interactive Exploration of Community Structure

The 2D plane of the network layout can be used as a new interaction domain for visual exploration. Interactive exploration can not only improve the network layout with inputs from users, but also provide users comprehensive information on the effects of combinations of eigenvectors. Based on the network layout algorithm, we allow users to decrease or increase the scales $w(x_i)$ for each of the k eigenvectors x_i. The effect of the interaction is the shift of one or multiple communities in the interaction domain, which are easy to identify during interactive exploration. This is important to compensate the loss of information when projection a k-dimensional space to 2D plane.



Step1: Initial layout

Step2:$w(x2) \downarrow$

Step3:$w(x1) \uparrow$, $w(x47) \uparrow$

Step4: $w(x48) \uparrow$, $w(x50) \uparrow$

Figure 30: Interactive Exploration of Community Structure. Positive edges are rendered in red and negative edges are in black. The labels show the adjustments of the scale values of individual eigenvectors.

Figure 30 demonstrates the process of interactive exploration. The curve of eigenvalues of this network has been shown in Figure 29. We first select the dimensions with the outstanding eigenvalues from the eigenvalue curve and generate a layout labeled by step 1. We further adjust the scales of eigenvectors manually and find additional community structures from step 2 to step 4. the changes of scale values are shown under each step. From our final layout, we can observe strong internal negative relationships in the middle four communities and positive relationships in the side two communities, which are approximate to k-block structure. Also, some external negative edges between communities indicate the network combines k-block and k-partite structures.

4.3.4 Discovery of Sub-Communities

The discovery of sub-communities can also be performed jointly with the interactive exploration of community structure. The main difference is that we gradually separate the communities. For every division, if interested, we treat it as a new network and visualize it until the remaining networks are similar to one of the two signature networks.

Figure 31 shows an example of finding sub-communities. The network contains 4 communities and 400 nodes totally. The internal edges are sparse and external edges are dense. As shown in the initial network layout, the k-partite network structure dominates the spectral space. We separate the three communities and visualize them as individual networks. Only one outstanding eigenvalue is shown for the two communities on the right, indicating there are no sub-communities. The network layout of the left community appears as a k-block network with two communities similar to k-partite networks. the curve of eigenvalues shows 4 outstanding eigenvalues. We separate the network to two communities and visualize them as individual networks. Finally, the two network layouts on the bottom show clear patterns: the network on

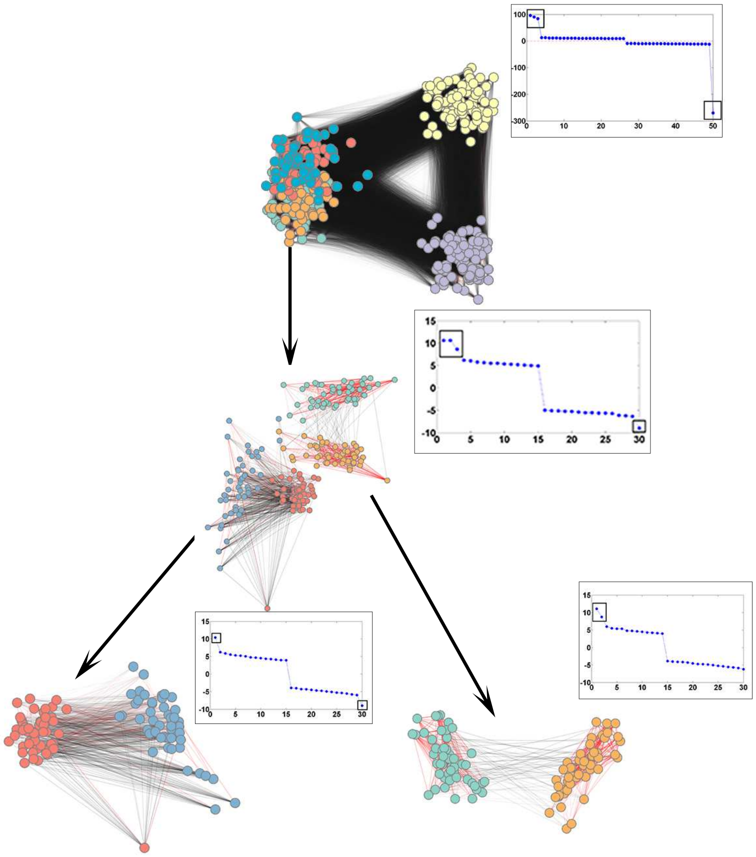


Figure 31: Interactive exploration of sub-communities.

the left is approximate to the k-partite network with many external negative edges between the two sub-communities; the network on the right is a mixed k-block and k-partite structure with dominant internal positive edges and a number of external negative edges.

4.4 Results and Discussion

4.4.1 Results

The example results provided in previous sections, such as Figures 26, 30 and 31, have shown our approach on a number of different networks. This section presents additional four results shown in Figure 32.

The first dataset is the Correlates of War dataset [27]. We filter the network by the years from 1993 to 2001, as the relationships of countries are relatively stable during this period. We mark the alliance relationship as 1, disputation as -1, and 0 otherwise. Our result of the network layout shows five major communities and a set of nodes loosely connected in the data (rendered in black). We can observe strong internal positive relationships approximate to k-block networks in the top four communities and strong external negative relationships approximate to k-partite networks between the community on the bottom and the other communities. Also, the three communities on the bottom demonstrate the combination of k-block and k-partite networks.

The second dataset is the network of American College football games between Division IA colleges during regular season Fall 2000 [53]. There are 115 teams and 12 communities, including 3 independent communities (dark green, grey, and pink) which have more connections with other communities than its own members. Our network layout visualizes the 8 major communities. Three communities rendered in dark green, grey and black, locate closely as they are densely connected to each other.

The third dataset is a complex synthetic network with 6 communities and 600 nodes. There are totally 13200 positive edges, 5970 negative edges, and 5000 random edges (can be either positive or negative) to increase the network complexity. The two outstanding eigenvalues correspond to two higher-level communities with 3 communities each. The layout of the third network shows that it appears as a k-block network with strong internal relationships. However, there are also strong external negative relationships among the three communities showing the triangle pattern on the top.

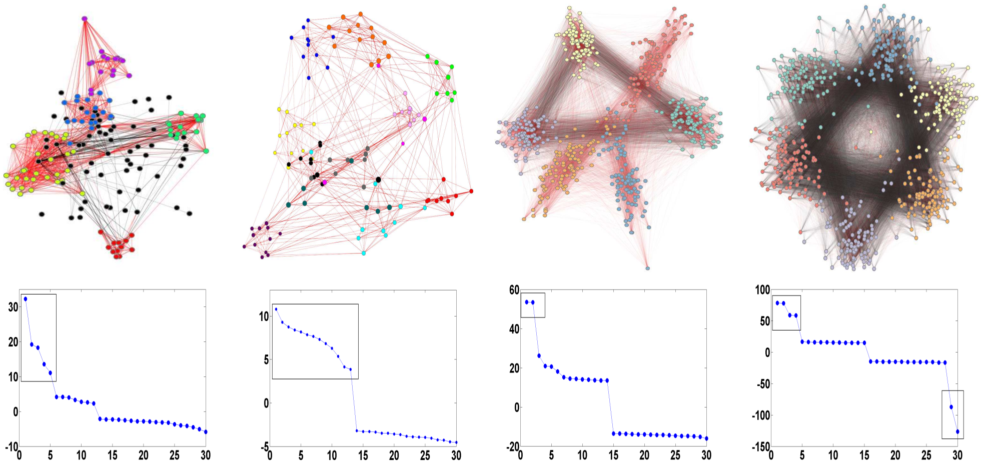


Figure 32: Examples of general signed networks. Positive edges are rendered in red and negative edges in black.

The layout also indicates internal positive relationships among these communities.

The fourth dataset is a synthetic network with 6 communities and 600 nodes. This network is generated by mixing with two approximate k-partite networks(one includes green, yellow and purple nodes, another includes blue, red and orange nodes). There are totally 3300 positive edges, 36000 negative edges, and 5000 random edges. The layout of the fourth network shows two overlapping triangle patterns, indicating the k-partite networks with strong negative external relationships. The two k-partite networks overlap on the 2D space due to a significant number of edges among the six communities.

4.4.2 Computation Complexity

The spectral decomposition is performed with QR algorithm by a reduction to Hessenberg form [126], which is $O(N^2)$ with N as the number of nodes in the network. In our experiment, this step only takes 0.1-0.2 second for networks with up to 1222 nodes and 33428 edges. The selection of representative subspaces involves hierarchical clustering, which is generally between exhaustive search in $O(N^2)$ to $O(N^3)$. In our experiment, this step can take several minutes for large networks. As we have the parameter k as the input, we can further explore acceleration techniques. The generation of network layout and interactive exploration are both $O(N)$. This is a nice feature as having a real-time interactive exploration process is crucial to visual analysis.

4.4.3 About community number k

Finding the number of communities k has been a challenging problem for several research areas. For networks similar to the two signature signed networks, our spectral analysis framework has provided a clear mechanism to identify them. For complex networks, such as networks with both densely k-partite and k-block structures, our spectral analysis framework can still reveals the major communities. It is important to understand that the number of eigenvalues selected does not necessarily to be the exact k.

Figures 30, 31, and 32 provide examples of general signed networks. In these examples, the numbers of outstanding eigenvalues do not always the same as the actual number of communities. This maybe caused by communities with different sizes or different edge densities. However, the curve of eigenvalues provides useful information for us to select the initial value of k and set up the search range.

4.4.4 About Community Structure

It worths to mention that spectral analysis studies the most significant data features in the spectral space. Many spectral analysis approaches are designed for balanced networks [19], in which the sizes of communities are comparable to each other. Otherwise, small communities are very likely to be treated as a sub-group of large communities. Without prior knowledge of community sizes, we should be aware that the order and dimensions of communities found in the study are not important. In this sense, our approach is capable of handling networks with unbalanced structures.

4.4.5 About Network Layout Design

The existing network layout algorithms have concentrated on unsigned networks [81]. Many of them do not take signed networks as input or do not generate correct results for signed networks. Force-directed layout algorithms can be used for signed networks. However, force-directed algorithms are often computational expensive, as the algorithm complexities are $O(n^3)$ [47]. Even for accelerated force-directed algorithms, the number of iteration is a problem [62].

Our visualization approach tightly integrates the network layout algorithm and the interactive exploration process. It is non-iterative and does not need much adjustment of parameters. The network layout is based on our spectral analysis framework and the visualization provides an interaction mechanism for exploring effects of individual eigenvectors without visualizing multiple spectral subspaces.

4.5 Conclusions

This chapter presents a study of signed networks from both theoretical and practical aspects. On the theoretical aspect, we have demonstrated and proved the relationships of spectral decomposition and community structures of signed networks. On the practical aspect, we have presented visualization and interaction methods to study signed networks, including a spectral-based network layout and interaction domain to identify important spectral sub-spaces and explore community structures. We have demonstrated with examples that our approach can successfully analyze community structures of different network types.

In the future, we plan to extend our approach to study large-scale social networks, including both signed and unsigned networks. We believe that spectrum-based approaches can provide effective and efficient visual analysis mechanisms to explore real and complex networks.

CHAPTER 5: NETWORK TOPOLOGY PATTERN VISUALIZATION

5.1 Introduction

With the wide adoption of wireless networks in real-life applications, enforcing security in these environments has become a top priority. We present an approach that can automatically suggest interesting data patterns. In all the intrusion detection mechanisms, several methods have been developed to visualize topologies using graph drawing or matrix representation [3, 79], since topology data is commonly collected in network applications and is extremely important for routing. A global network topology records the neighbor relationships among wireless nodes and includes many traces that can be used to detect attacks on authentication and node identities.

With the ever increasing data size and complexity, many visualization approaches have been developed to improve the processing of a large amount of network data including traffic patterns, network flows and logs [10, 96, 140]. Because of the importance of the network topology, it has been used to help enforce Internet and wireless network security in multiple network visualization mechanisms [7, 72, 129]. For example, topologies have been visualized using graph drawing or parallel coordinates [1, 3] to show interesting patterns of malicious attacks.

Specifically, we investigate *Sybil attack* [36] that manipulates node identities under various attack scenarios in wireless networks. In such attacks, a single malicious node plays the roles of multiple legitimate members of the network by impersonating their identities or claiming fake IDs. Since in these attacks the malicious nodes can change the number of fake identities and their connection relationships freely, the effectiveness of previous intrusion detection systems may be drastically weakened [36]. More details about the behaviors and impacts of Sybil attacks are provided later. Noticing the serious harm that Sybil attacks can cause, researchers have proposed several approaches to defend against such attacks. Existing approaches usually concentrate on verifying whether or not a pair of nodes have distinct resources, distinct knowledge or distinct positions. However, these automatic algorithms often make certain assumptions about the environments and are not capable of detecting complex variations of Sybil attacks. We believe that visualization of global topology is one promising direction.

In this chapter, we reorder the time-variant network topology and extract special patterns of Sybil attacks for their detections. We believe that the proposed techniques to model patterns of attacks can be applied to the detection of a broader range of malicious activities.

5.2 Sybil Attack Detection

Sybil attack is one particularly harmful attack on distributed systems [20] and wireless networks [36]. This attack has been demonstrated to be detrimental to many important network functions. For example, the Sybil attack is discussed in an architecture for secure resource peering in an Internet-scale computing infrastructure [48]. Newsome et al. [100] have also pointed out that combinations of different types of Sybil attacks may cause severe impacts on wireless sensor networks, which are very difficult to recover.

Existing detection methods can be divided into two categories: identity-based or location-based approaches. The first category generally mitigates Sybil attacks by limiting the generation of valid node information, such as the approach of pre-distributed secret keys [100]. The second category utilizes the fact that each node can only be at one position at any moment, such as the SeRLoc approach that determines node locations passively under known attacks [89].

Now, we briefly describe the behaviors of Sybil attacks and their potential harm to a wireless network. As the name of Sybil attacks implies, malicious nodes play the roles of multiple legitimate members in a network by impersonating their identities or claiming fake IDs. These fake nodes do not have real physical devices like legitimate nodes and they often claim to have direct or indirect connections with the malicious nodes that generate them. Here, we borrow the taxonomy defined in and classify the attacks based on the connections among Sybil and legitimate nodes. If the Sybil nodes can directly communicate with other legitimate nodes, it is a direct Sybil attack. In contrast, in an indirect Sybil attack, a malicious device claims to have the paths to reach Sybil nodes so all messages have to go through it. Although Sybil attacks seem to be simple, they can affect network performance at different degrees and cause severe harm, such as manipulating the results of localized voting or data aggregation. In the worst case, Sybil attacks can enable malicious nodes to take over the control of the whole network and defeat the replication mechanisms in distributed systems.

The main difficulties in detecting Sybil attacks come from various combinations of individual attacks. Although it is difficult to link together multiple fake identities that appear in different periods of a network's lifetime and detect non-simultaneous attacks, their impacts on network security are also limited. For example, a Sybil node that is not a member of a network cannot cast a vote during the leader election procedure. Therefore, in this article, we focus on the simultaneous Sybil attacks. To evaluate our proposed mechanism in a more realistic environment, we assume that both direct and indirect attacks exist in the network and a malicious node can dynamically switch between the two types. We also assume that multiple malicious physical devices coexist in the network and a Sybil node can switch among them.

5.3 Network Topology Pattern Visualization

Our detection approach is achieved through pattern generation. This section presents our pattern generation methods based on attack features. We first describe how we collect network topology information from wireless nodes. Then, we present the topology patterns that can be used as indications of Sybil attack existence.

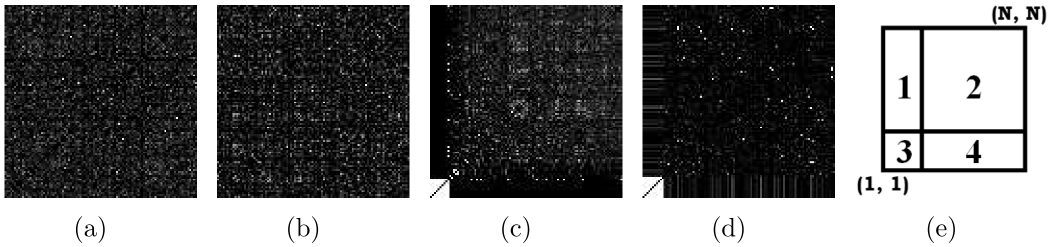


Figure 33: (a,b) General 2D statistical topology matrices do not reveal any suspicious patterns; (c) Signature pattern for indirect Sybil attacks; (d) Signature pattern for direct Sybil attacks; (e) A 2 by 2 grid structure in the patterns, with index (1, 1) at the left bottom corner.

5.3.1 Global Network Topology Patterns

Since Sybil attacks do not demonstrate anomalies in neighbor relationships at individual time steps, we need to collect the connectivity information among wireless nodes for a time period to detect Sybil nodes. Assume there are N nodes in the network and the time range is $[0, R]$. At each sampled time step, the connectivity relationship can be represented as an $N \times N$ topology matrix T, with $T(i, j)$=1 indicating the connection between node i and node j. In this way, the information of network topology across a time period can be represented as a 3D $N \times N \times R$ connectivity matrix. We can use central controllers that are special nodes in a wireless network to collect network topology information. We summarize the 3D connectivity matrix into a 2D global topology table, which records the number of time steps that each pair of nodes are connected during the time period under study. We choose to concentrate on analyzing 2D global topology patterns, since they are convenient for users to visualize. The signature patterns of Sybil attacks are found to be 2 by 2 grid structures, as shown in Figure 33. Generally, a topology pattern across any time period appears to be random without special arrangements (Figures 33 (a) and (b)). When we reorder the node sequence, we may see some interesting 2 by 2 grid structure patterns, as shown in Figures 33 (c) and (d). These two special patterns are closely related to the attack procedures and indicate the existence of malicious nodes. Simply speaking, Sybil attacks can be summarized as a malicious device presenting multiple identities to the network. There are two types of Sybil attacks: direct attacks, in which malicious nodes use multiple fake identities to directly communicate with other nodes; and indirect attacks, in which a malicious device claims to have the paths to reach the Sybil nodes and all messages have to go through it. Because of the time and location constraints, similar signature patterns are exposed when malicious nodes and legitimate nodes are separated in the 2D statistical matrix. According to the pattern features, we have developed several automatic arrangement methods to expose patterns that are similar to the signature patterns. To illustrate our pattern generation algorithms, we divide topology patterns into 2 by 2 grid structures, as shown in Figure 33 (e).

5.3.2 Pattern Generation

We design automatic algorithms to expose the patterns hidden in the global topology matrix that are similar to the signature patterns of Sybil attacks. Our approach is to generate new patterns by reordering node sequences along the two dimensions of the global topology matrix. Since the 2D topology matrix of a network with N nodes may generate $N! \times N!$ different patterns, it is obviously too time consuming for users, such as network administrators, to manually adjust node sequences. Therefore, we need to automatically arrange node sequences during the decision making process, especially for complex attack scenarios and large scale networks.

We use the features of Sybil attacks to guide our automatic pattern generation processes. We have analyzed these attacks from multiple aspects and designed matrix reordering algorithms according to each attack feature. These patterns are then automatically evaluated and organized for users to detect attacks interactively. Generally, we can declare the existence of attacks as long as one of the reordered sequences shows a suspicious pattern. This approach allows us to analyze the 2D global topology matrix from multiple independent or correlated aspects. We show in our results that this method is convenient and robust for detecting various Sybil attack combinations. The following describes three automatic pattern generation methods that have been found to be effective in detecting our signature patterns.

Method 1 - Anchor Connection: Our first method is designed for indirect attacks according to the connection feature of Sybil nodes and legitimate nodes. As shown in the signature pattern of indirect attacks in Figure 33 (c), there are two blocks that are almost empty: regions 1 and 4. This statistical feature is a result of the lack of direct connectivity between Sybil and legitimate nodes and long-time connections among the Sybil nodes. Corresponding to the attack definition, indirect Sybil nodes can communicate with legitimate nodes only through a small number of malicious 'anchor' nodes. Although this may not be obvious at a single time step, it becomes more and more visible in the statistical matrix with the increasing length of the monitored time duration. Our first method is designed to reorder the global topology matrix to form such patterns through the following procedure:

Step 1: For each row, measure its connectivity degree by accumulating the square of data values;

Step 2: Sort rows in decreasing order of connectivity degrees;

Step 3: Apply the row sequence to the column.

Since the initial global topology matrix is symmetric, we can apply the row sequence to the column directly to sort the connectivity degrees. As shown in the second row in Figure 34, this method successfully captures this feature of indirect attacks.

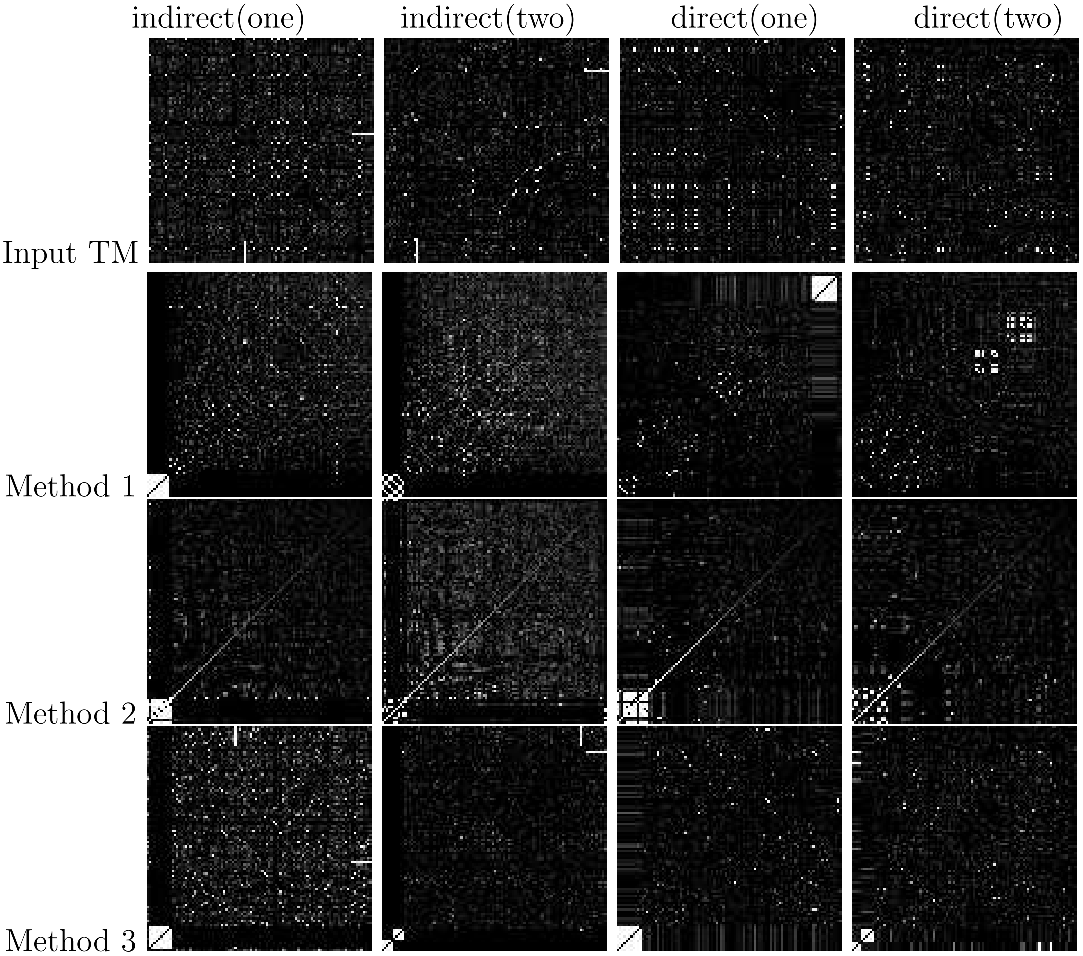


Figure 34: Examples of our pattern generation results. The first row shows four 2D statistical topology matrices and the second to fourth rows show their corresponding reordered patterns from methods 1 to 3. These four datasets contain one group of indirect attack nodes, two groups of indirect attack nodes, one group of direct attack nodes, and two groups of direct attack nodes respectively. Most reordered topology matrices demonstrate more obvious attack patterns than their initial matrices.

Method 2 - High Connectivity: Our second method is designed for both types of Sybil attacks according to the high connectivity feature among fake identities. As shown in Figures 33 (c) and (d), the left bottom corner (region 3) of our signature patterns accumulates a block of bright pixels. This indicates the existence of a group of highly connected nodes in the network. Corresponding to the nature of these attacks, since multiple Sybil nodes are fabricated by the same physical device, their locations are usually close to each other. These malicious nodes often have to claim that they are connected to avoid being detected by location-based methods. We design this method to form a large value block at the left bottom corner in a global topology matrix through the following procedure:

Step 1: Repeat the steps 2-4 from $m = N$ to $m = 2$ to reorder the whole pattern.

Step 2: Scan the top right $m \times m$ region and select one item (i, j) with the largest value;

Step 3: Switch row $N - m + 1$ and row j;

Step 4: Switch column $N - m + 1$ and column i;

This feature is especially useful in detecting direct attacks since the fake identities communicate directly with legitimate nodes, who will report all the connections honestly. Even if attackers increase the number of Sybil nodes to reduce their average connection number, they still need to keep high adjacency values for the attack effectiveness. The row titled 'Method 2' in Figure 34 shows that this method is useful for the detection of both direct and indirect attacks. Although these patterns may not be as obvious as our signature patterns, they are clear enough for users to capture this feature.

Method 3 - Close Locations: Our third method is designed for both types of Sybil attacks according to the moving feature of Sybil nodes. As shown in the signature pattern of direct attacks in Figure 33 (d), regions 1 and 4 demonstrate clear horizontal and vertical band patterns. Actually, the empty regions 1 and 4 of indirect attacks in Figure 33 (c) can also be viewed as a special case of these band patterns. Corresponding to the attacks, this feature indicates similar movement patterns of Sybil node groups, while legitimate nodes rarely share the same moving trace for a long time duration. This is inevitable due to the fact that Sybil nodes are attached to the same physical device. At a single time step, we can input the topology matrix to a multi-dimensional scaling (MDS) method [125] to reconstruct the physical distances among the wireless nodes in a network. Similarly, the reconstructed locations from a statistical global topology matrix can be used to measure average node distances in a time duration. With the similar moving patterns of Sybil nodes, we can use the distribution of reconstructed locations to separate malicious nodes from legitimate ones. Figure 35 shows two examples of malicious nodes separated from the legitimate nodes. We design this method to group the nodes based on their reconstructed locations from the statistical matrix:

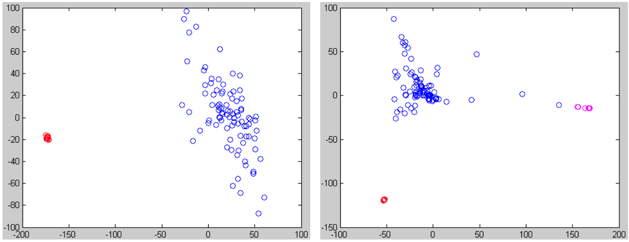


Figure 35: MDS reconstructed node locations can be used to detect Sybil attacks, since malicious nodes tend to move in groups during a time period. Legitimate nodes are colored blue and two malicious groups are colored red and purple, respectively.

Step 1: Calculate a dis-connectivity matrix by reversing the global topology matrix: $D(i, j) = 1 - T(i, j)$;

Step 2: Reconstruct 2D statistical node locations using MDS method;

Step 3: Calculate the center position of all the nodes;

Step 4: Reorder the sequence of nodes according to their distances to the center position in a decreasing order;

Step 5: Apply the sequence to both row and column.

The row titled ‘Method 3’ in Figure 34 shows that this method is useful for both direct and indirect attacks. We expect that clustering algorithms can be applied to improve this algorithm.

5.4 Discussion

The design of multi-matrix visualizations plays the major role in enabling the capability of detecting complex Sybil attacks. Matrix visualization allows users to detect subtle patterns using visual cues and user expertise. While it is natural to use a matrix visualization approach to visualize topology information, other information visualization methods, such as parallel coordinates, can be used as well. We can use the row and column sequences as the orders of two axes in parallel coordinates to show highlighted band patterns among malicious nodes. As shown in Figure 36, we use the same scheme to color the matrices and parallel coordinates for comparison. Since we visualize statistical topology matrices, the node connections are dense, which are more challenging for parallel coordinates because of the line overlapping issue. On the left column, both visualizations do not show obvious patterns; while on the right column, both visualizations represent the highlighted region well, but parallel coordinates do not have the band structures in the matrix visualization. Also, since topology data is represented as matrices, matrix visualization is more intuitive for users, who are likely familiar with the matrix representation of topologies, to understand; thereby easier for them to identify malicious nodes. Therefore, we believe that matrix visualization is more appropriate for this application.

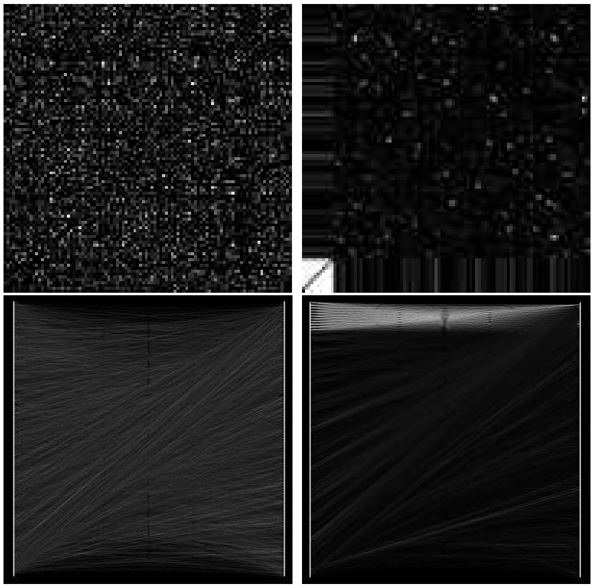


Figure 36: Comparison of matrix visualization and parallel coordinates. Each column visualizes a topology matrix with the same color scheme.

5.5 Conclusion

We present a robust approach to detect Sybil attacks in wireless networks through analyzing statistical topology patterns. We characterize the attack features and detect malicious nodes with automatic pattern generation methods. Since we consider multiple relevant topology patterns, our method is robust to the detection of various attacks according to different aspects of attack features.

CHAPTER 6: COLLABORATIVE NETWORK VISUALIZATION

6.1 Introduction

Collaborative analysis can benefit many large scale applications where a small group of users discuss and negotiate their interpretations of the data with which they are working. These techniques are often required for application fields where task complexities can easily overrun computing power and algorithm intelligence [58, 65, 74]. Especially in security applications, collaboration mechanisms are crucial to provide time efficient solutions for processing a large amount of data in real time. While approaches have been explored for assisting individual analyst with detection tasks, the problem of collaborative analysis for such applications is still open.

In this chapter, we concentrate on exploring a suitable solution for complex intrusion detection applications. Our motivation comes from the fact that important networking environments are always protected by security teams. Traditionally, such teams would use an algorithm-based Intrusion Detection System (IDS) to warn them of threats, but IDSes are prone to give false alarms. Since every alarm must be verified, security teams end up wasting much time investigating events that turn out to be false alarms. Collaborative analysis can provide a practical solution to overcome the ineffectiveness of automatic detection algorithms, limits of computing resources, and complexity of advanced malicious attacks. Especially for intrusion detection, where new or unknown attacks are often introduced, having a group of experts analyzing data in real time is crucial to provide accurate and time critical results.

Since most previous detection approaches have been designed from a single-user perspective, it is usually not possible to apply them directly to team coordination. Relevant studies, such as ones exploring teamwork theories, have been extensively performed in the fields of artificial intelligence and robotics. However, the proactive collaborative problem-solving feature of the security domain differentiates it from multi-agent coordination in their applications. To build an effective collaborative visualization model, various aspects related to the collaborative problem solving process, such as knowledge sharing and social factors [93], should be considered. However, it is not yet understood how interfaces and interaction techniques should be designed to specifically address the needs of distributed collaborative analysis.

In this chapter, we design and develop a collaborative detection mechanism for defending against complex malicious attacks in wireless networks. The goal of our collaboration teams is to identify hidden attacks and remove malicious nodes from the network. Here we concentrate on defending against a particularly harmful attack known as the Sybil attack, which has numerous variations. We first analyze the requirements for designing such coordinated systems. Later, we describe a web-based prototype system, which is built based on our design principles and heuristics. Our system supports multi-user input, shared and individual views on detection findings, and flexible workspace organization to facilitate group analysis.

The main contribution is that our work explores the area of collaborative analysis in a distributed setting, which to our knowledge has not been explored in significant depth. Our approach incorporates results from several research fields, including models of human behavior, teamwork theory, and interface design. We also provide a detailed discussion of different collaboration aspects. We give practical solutions for security applications in real-life for defending against various attacks, assuming that a reasonable detection algorithm is provided for each representative attack. The web-based prototype system and networking data collections provide a testbed for other researchers to explore and evaluate the effectiveness of different coordination aspects, which are hard to access without a working example.

6.2 Design Guidelines and Heuristics

This section describes several guidelines for assisting a small team (3-15 members) on monitoring and defending a network collaboratively with a distributed environment. Our design references knowledge from multiple fields, including social science, psychology collaboration models and teamwork theory; and our observations of team dynamics in security applications.

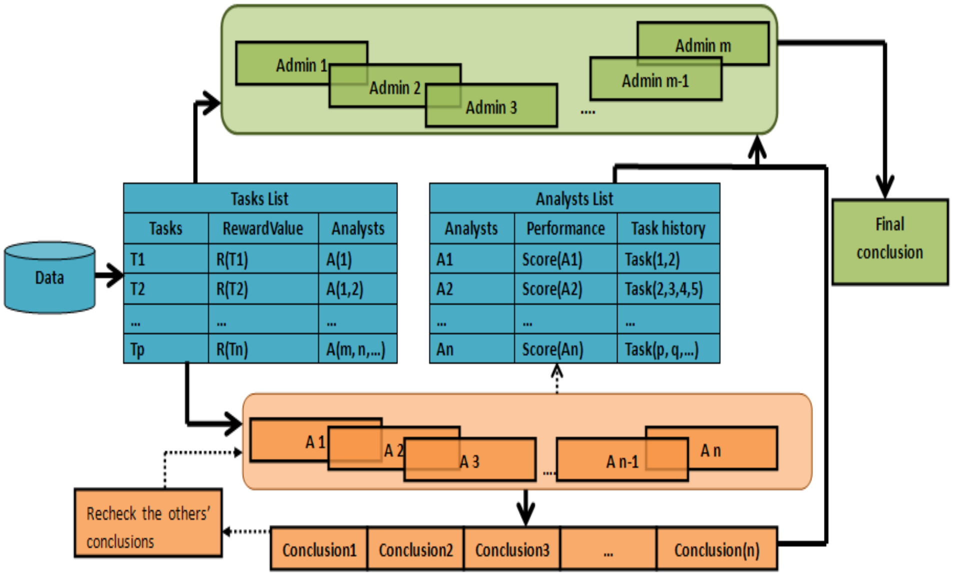
6.2.1 Designing the Roles

The structure of participant roles can directly affect the efficiency of a collaborative problem-solving team. In real applications, the privilege of making final decisions can only be given to a small number of participants; thus we need to separate the roles of participants into at least two groups: *administrators* who supervise the collaboration process and *analysts* who handle individual detection tasks.

In our design, analysts are treated as the main players in the collaborative analysis process. Analysts can actively choose their tasks and coordinate with each other to complete the detection process. On the other hand, administrators have the responsibilities of monitoring team progress such as reviewing the results from analysts, adjusting task rewards, adding new tasks to the task list, monitoring analyst performances, drawing final conclusions, and removing malicious nodes from the network.

6.2.2 Designing the Workflow

Workflow is a powerful tool to guide collaboration and monitor overall team performance [33]. Our efforts are focused on designing a mechanism to smooth the coordination and communication among analysts.



Tasks List

Tasks	RewardValue	Analysts
T1	R(T1)	A(1)
T2	R(T2)	A(1,2)
...	...	...
Tp	R(Tn)	A(m,n,...)

Analysts List

Analysts	Performance	Task history
A1	Score(A1)	Task(1,2)
A2	Score(A2)	Task(2,3,4,5)
...	...	...
An	Score(An)	Task(p,q,...)

Figure 37: The workflow of coordinated detection. A list of analysis tasks is generated in real-time at the server and can be accessed by both administrators and analysts. Analysts handle the detailed detection process, while administrators overview the team performance and make final decisions on action. The workflow provides administrators with the flexibility to monitor and adjust the team progress and analysts capabilities of collaborative analysis.

As Figure 37 shows, the workflow starts from a server, which collects and stores data from the network in real time. For generality, we only collect network topology, which is one of the most commonly used information sources in network security. Once the detection process starts, the server automatically generates a list of tasks by dividing data into equal time durations. We define the set of tasks as $T := T_1, ..., T_p$, where p is the number of tasks defined in the system. Each task has an associated estimate cost of time and a reward value $R(T_i)$, which changes with time and detection results to promote early selection of important tasks.

In the workflow, administrators can access all the data and findings from individual analysts. The administrators are the decision makers who review findings from analysts and draw final conclusions by considering the whole detection process. They are also responsible for monitoring overall team performance and improving the team efficiency by actively assigning tasks to analysts, modifying reward values, and suggesting the involvement of additional analysts.

Analysts must detect and explore hidden attacks by studying various matrix patterns generated from different time periods and investigate suspicious activity. Analysts first either select a task from the generated task list by considering its reward value or find that they have been assigned a set of tasks by an administrator. At this point, the analysts explore this assigned task via the provided visualization and interaction tools. During this analytical process, sharing information is needed. Analysts can share their conclusions and suggestions through images or text stored in a repository on the server. Thus, other analysts can reference historical findings for the task in order to make a more comprehensive decision regarding suspicious activity. Finally, the server automatically updates the task list and reward values based on the analysts' conclusions.

6.2.3 Designing Collaborative Detection

The following describes the interaction and collaboration regarding three aspects: detection, coordination, and communication.

6.2.3.1 Detection

Our design of collaborative analysis is applicable to general intrusion detection tasks. We concentrate on detecting Sybil attacks by topology matrix visualization. The details of detection strategies and visualization can be found in Chapter 5.

6.2.3.2 Coordination

As described in [82], "coordination is the attempt by multiple entities to act in concert in order to achieve a common goal by carrying out a script/plan they all understand." Thus building up such a good "script" for team members is the key point for the whole collaboration process. For distributed collaborative environment, Neale et al. [98] have pointed out that coordination should be defined by the combination of procedures, tasks, tools, and communication. In this section, we discuss the procedures of coordination including the division and allocation of task, the updating of reward values and performance scores, and decision making. In section 6.2.3.3, we discuss communication.

Collaboration style: We define two stages in the detection process: detection and monitor stages. The detection stage is the duration from the announcement of the first suspicious node to the time step in which all known attackers have been removed. The monitoring stage is time during which there are no obvious attacks occuring. In our distributed security environment, collaborators switch their collaboration styles between types of loose style in monitor stages and close style in detection stages. Collaboration style is also related to the division of tasks.

Task reward: In order to support efficient collaboration among the analysts, providing a guide for them to target at time-critical tasks is necessary. Therefore, we design a reward value metric that is associated with each task. All the tasks are initially assigned reward values of 0s. The reward values are updated according to analysts' findings of suspicious nodes. Its value increases by 1 when any new or additional suspicious nodes are found, or conflicting conclusions for the same tasks are drawn.

Therefore, task reward value is the sum of the number of suspicious nodes and the times of all the conflict results. If the reward value is high, it indicates the task is in high risk and needs to remove the malicious nodes as soon as possible, or the task is too complex to make a clear conclusion by only two or three analysts, it needs more analysts to double check. Administrators can use reward value to control the task allocation. They can assign the task with high value to more analysts as well.

Division of task: Division of tasks is one of the most basic aspects in collaboration work. However, it is not trivial to parallelize an entire workflow into proper independent units [65]. The data we use in our system is temporal network data, so we divide tasks based on time duration.

The choice of task duration $d(t)$ at the time step t can be adjusted through two factors: the response duration of analysts and their collaboration styles. We require that each task be engaged by several analysts simultaneously. Thus response duration is the time costs between the selection and conclusion of a task from all the analysts for all the tasks during a recent history. We calculate the average $ar(t)$ of these response durations. Generally, a Sybil attack can cause severe damage within a certain time TD_{max}. The maximum change is defined as D_{max}. Thus, we design an equation 14 for division of tasks. The second factor is collaboration style, meaning that a longer duration for monitoring stages and a shorter duration for detection stages when communication is more frequently needed. We use half d for the detection stage.

$$d(t) = \begin{bmatrix} -D_{max} \times (\frac{2ar(t)}{TD_{max}} - 1)^3, if\ TD_{max} >= ar(t) >= 0 \\ -D_{max}, if\ ar(t) > TD_{max} \end{bmatrix} \tag{14}$$

Allocation of task: Effective division of task is not sufficient for successful collaboration, the efficient allocation of tasks is also necessary. In order to allocate tasks to proper individuals, analysts in our design can choose new tasks on their own with the information of reward values and their own task history. Allowing analysts to manage their work independently can bring benefits, in contrast to assigning their task passively [42]. The latter approach requires a central planner to control each analyst's workload, and thus the central planner must know much precise information about the whole network state and the analysts' respective productivity capacities. In such a management structure, even small mistakes made by the planner can drastically affect the entire organization. Allowing analysts to actively select tasks avoids this problem. Additionally, every analyst can concentrate on his or her own tasks without concerning about what tasks the other analysts are working on. Furthermore, administrators in this model are able to adjust and take action on the incoming results from the analysts. This model may slightly reduce the output of the analysts, since they must take time to select their own tasks. However, we believe that the time gained by administrators and the benefits brought by analysts having a say in their task allocation outweighs this small time loss for the analysts.

Task coordination: An important aspect of collaboration among analysts is referring to the regions deemed suspicious by the other group members through a spatial context [65]. Clark [25] grouped many forms of spatial reference into two categories: pointing and placing. Pointing means that using some vectorial reference to direct attention to specified regions or objects. Placing means that moving some information into one shared-space. In our system, we provide a task list and the corresponding suspicious node list to promote coordination among all participants. Analysts can point out their discoveries of suspicious nodes or regions in generated images after detection. Afterwards, they can upload their conclusions to the server, and others can verify these conclusions. Analysts test conclusions from others by reordering topology patterns according to their assumption of suspicious nodes.

Performance measurement: As described by Shipman and Wholey [116], “Performance measurement is the ongoing monitoring and reporting of program accomplishments, particularly progress towards pre-established goals”. In our design, we build quantitative performance standards to improve the accountability of each analyst and the general effectiveness of coordination. To do this, we collect correctness and performance scores for each analyst. Correctness can be measured by comparing the analysts’ conclusions for each task with the administrators’. The percentage of correctness increases when the conclusions are identical, and otherwise decreases. Thus the suspiciousness degree of any given task can be modified by administrators according to the correctness scores of the analysts who processed the task's data. Likewise, we measure the performance score by accumulating all the final reward values of the tasks an analyst has processed. We can assess productivity of each analyst by this score. The performance list is only available to administrators.

Decision Making: Decision making is a comprehensive procedure. In our approach, administrators can make final decisions about action by examining analysts' results, the task reward values and analysts' performance scores.

The administrators do not need to know the details of each analyst's work, but the system allows them to change task reward values and assign tasks to analysts when they can not make final conclusions by unclear sources such as the tasks contain uncertain suspicious nodes.

6.2.3.3 Communication

Sharing information: Sharing information is important in collaborative work groups [30]. Brennan et al. [41] built a collaborative framework among multiple analysts. In this framework, they focused on the idea of common grounded [24] communication, which allowed multiple analysts to share information, especially the reasoning behind the information, logically and graphically. Sharing information in our system is based on this framework. Analysts can point out their findings in generated images and send their findings with conclusion and suggestion to a sharing space in server. They can update the lists of malicious nodes information with confidence values. They are also permitted to write notes based on their own authority and expertise.

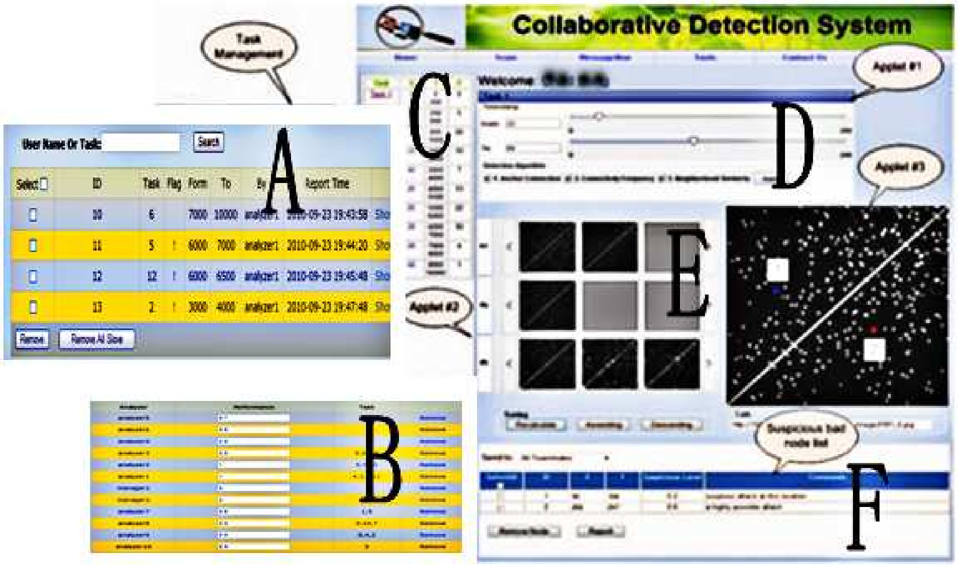
Awareness: One important aspect of communication is to provide the work status of each team member to the others. For distributed work groups, it is difficult to maintain awareness of the other members' work status [49] because of geographical distance. The traditional ways of maintaining awareness in distributed work groups (such as email) were demonstrated to be inefficient [85]. To mitigate this, we design a new way for analysts to maintain situational awareness. In our system, the ongoing task list and the analysts working on tasks are provided to further enhance awareness. Analysts can view the examined and unexamined tasks and the work progress of other analysts.

6.3 Collaborative Detection System

We apply the above design guidelines to develop a prototype system. We choose a web-based solution, as it is convenient for a group of people to monitor and defend a network collaboratively in a distributed environment. That is, through the web-based collaborative platform, multiple network analysts and administrators can work collaboratively towards identifying suspicious network events. Intelligent control mechanisms are also used for user management, task management, and collaborative decision making. The following describes the interface and implementation of such a web-based tool.

6.3.1 Interface Design

Figure 38 demonstrates the user interface design of the web-based collaborative detection system. The functionalities supported by the interface can be categorized



Select☐	ID	Task	Flag	Form	To	By	Report Time	
☐	10	6		7000	10000	analyzer1	2010-09-23 19:43:58	Sho
☐	11	5	!	6000	7000	analyzer1	2010-09-23 19:44:20	Sho
☐	12	12	!	6000	6500	analyzer1	2010-09-23 19:45:48	Sho
☐	13	2	!	3000	4000	analyzer1	2010-09-23 19:47:48	Sho

Figure 38: A demonstration of the graphical user interface for collaborative detection. (A) working processing board. (B) performance list for administrators. (C) the task list. (D) a panel for selecting time ranges and detection algorithms. (E) topology pattern window. The topology patterns are arranged into three rows which are results from three algorithms. When analysts double click one pattern, an enlarge image will be shown in the right window. (F) suspicious nodes list which is generated automatically based on suspicious nodes identified by analysts. Analysts can identify suspicious level and write comments here.

into three types: (1) Administrative functionalities: For the purpose of effective user interaction, users are required to register and login before utilizing the functionalities supported by the system. An administrative interface is provided for administrators. Administrators can check the tasks detail information (Figure 38 (C)), report time and suspicious nodes list (Figure 38(F)) and the performance list (Figure 38 (B)). They make final decisions by considering all the information comprehensively. They can indicate the final conclusion of the tasks by adding 'Flag' in working processing board (Figure 38 (A)). If the 'Flag' is '!', it means that suspicious nodes have been found. They also can remove '!' if they decide the task is safe. (2) Visual analytics functionalities: The interface displays visual representation of abstract network data to a group of users. Chart controls are provided to accept necessary user interactions, from moving the mouse over a 2D location to clicking or double-clicking on that location, for marking suspicious nodes under attack. Analysts can also adjust sliders in Figure 38 (D) to select different time ranges of the detected task and to select different algorithms. This interaction make analysts to get the precise location of the suspicious nodes. In Figure 38 (E), the generated images by three different detection algorithms locate at different rows. Analysts can get a scaled image by double clicking the small one. In the scaled image, they can point suspicious nodes in red or good nodes in blue by clicking them. (3) Communication and coordination functionalities: The findings of each task can be shared among analysts. They can click 'report' button to upload suspicious node list with images and notes (Figure 38 (F)) to tell the other analysts the reasons of their findings. Our interface also helps both analyzers and administrators manage and maintain lists of ongoing tasks, and for each task, keep its allocation status, its current reward value, and a list of suspicious bad nodes.

6.3.2 Implementation Detail

Following the Model-View-Controller design pattern, for reasons of flexibility, the implementation of the interface supporting visual analytics functionalities consists of three modules: the Data module, the Control module, and the Visualization module. For each user request, these three interacting components always work together to produce visual representations of the network data in a user-specified range. Specifically, the Visualization module sends requests to the Control module for the display content while the Control module sends requests to the Data module for the network data that is required for satisfying the display requests.

6.4 Discussion

Here we discuss several different scenarios of attacks: no attack, simple attack, and complex attack. It is important that a distributed collaboration system is able to handle all the cases as each case has potential pitfalls.

In the case that there are no attacks in a network, all analysts should be made aware of this fact. In a traditional setting that relies on automatic intrusion detection systems, false alarms are common. As a result, many analysts are kept busy with verifying that false alarms are indeed false alarms. In our system we rely on the strength and diversity in abilities of a group of analysts to assess the state of a network. As such, when the network is in a safe state, our collaborative tools allow this fact to propagate to all analysts and administrators, who may then reduce the number of analysts actively working on tasks and let them either explore historical data or devote their time to other tasks.

The second case is where there is a simple attack. A potential pitfall in this situation, particularly when several analysts are processing the network data, is repeated work. That is, in a system that employs several analysts but fails to have adequate communication capabilities, several analysts may go through the process of identifying the simple attack. However, since our distributed system provides communicative functionality to both administrators and other analysts, the analysts who have not yet processed the attack will be made aware of the fact that an attack has been identified, and may then assist in verifying this conclusion. By verifying the conclusion, the analysts are updating the suspiciousness degree, which will prompt final action by an administrator more quickly than traditional communication methods.

Finally, the case of a complex attack is when attack nodes are migrating and exhibiting other complex behaviors. Such an attack may be first discovered by an analyst. However, it is more likely that the administrators, who are concerned with processing conclusions from the analysts, will identify patterns based on the results submitted by the analysts. At this point, administrators can produce and deploy a verification and response plan using the provided visualizations and tools. Our system gives administrators the ability to respond rationally (with the correct amount of analysts) and in a timely manner.

An additional example of a related complex scenario is when conflicts arise among different analysts' conclusions for a task. For some nodes of the task, different analysts may draw different conclusions based on their respective expertise. For example, analyst A finds a malicious node in some task's data. As a result, analyst A will increase the reward value of this task. Before A sends his results to the server, analyst B processes the same task. However, analyst B identifies the node that has been identified as malicious by A to be benign. As a result, this task is viewed as having conflicting results and the reward value is increased further. Our goal then becomes to eliminate the conflict. Administrators can identify such situations and assign more analysts to process this task while updating the respective performance and correctness scores of A and B. Based on the results from additional analysts and their respective performance scores, administrators will be able to draw conclusions about the task in question.

Another interesting situation arises when the reward value of the task is the same but different malicious nodes are found. To address this in our design, we provide a task list and corresponding suspicious node list with suspiciousness degrees to all participants. The administrator can assign nodes with high suspiciousness degrees as tasks to analysts for verification, and finally make decisions based on the analysts' conclusions.

6.5 Expert Feedback

As an important component of the evaluation procedure, we have provided the prototype system to four researchers: two visualization researchers, one wireless network security researcher and one web-based collaborative analysis researcher. The following summarizes the positive feedback from three aspects. Limitations and future work are summarized in the next section.

First, the feedback of the network security researcher shows that the hierarchical organization of the user roles matches the scenarios of many real-life applications. A system administrator often has several assistants in managing a complex wireless network. Once the distribution of the work responsibilities among the assistants is determined, they usually have a great degree of flexibility in accomplishing their tasks. At the same time, the administrators have the authority to integrate the results from the analysts and make the final decisions. One feature that distinguishes the proposed approach from several voting based attack detection schemes is that the analysts can choose their own tasks based on their expertise, processing capabilities, and rewards. It liberates the administrators from the overhead of task assignment so that they can focus more on the result integration procedure.

Second, the web-based collaborative analysis researcher comments that the proposed approach provides a powerful and convenient vehicle for communication among the analysts. The system provides two channels for the analysts to share their observations and localized detection results. First, they can identify the suspicious areas in the network so that other analysts can conduct detection at a finer granularity in the areas. Second, the analysts can directly share the suspicious that they identify and assist other analysts in their tasks. Sharing only the suspicious areas and the detection results will greatly reduce the communication overhead among the analysts.

Third, all the experts believe that the methods to measure the performance of analysts are very necessary for collaboration work. The schemes include the reward incentives and the performance monitoring procedures. The reward incentives inspire the rational analysts to carefully conduct the attack detection tasks to maximize their evaluation effectiveness. At the same time, the cross-comparison between the analysts' results and the final decisions of the administrators prevents the analysts from trading detection accuracy for response time. The two schemes together can reduce the false positive and false negative alarms. At the same time, the degradation of the performance of any analysts can be easily discovered by the administrators.

6.6 Limitations

We hope our work will bring a discussion on exploring collaborative visualization methods for information security applications, especially in distributed settings. There are still several limitations of our present work. One limitation is that the communication among distributed team members is limited. We plan to improve it for time-critical applications by allowing analysts to share ongoing results through introducing uncertainty visualization to our system. Another limitation is that it is still challenging for administrators to make decisions when conflict results occur. Thus additional decision making tools are necessary for such case. We plan to study relevant work from social science to improve the workflow for this purpose.

CHAPTER 7: EVALUATION OF COLLABORATIVE VISUALIZATION

7.1 Introduction

With the increasing challenges of data analysis and visualization, collaborative problem-solving has started to attract interest of visualization researchers. Recently, Bresciani and Eppler [17] analyzed the impact of visualization on knowledge sharing in situated work groups and showed that interactive visualization could bring positive and productive changes for group work. As described in [74], while the concept of collaborative visualization is not new, research on the process of collaboration is relatively scarce. To develop effective collaborative analysis solutions, we need to explore the benefits of coordination and communication for the appropriate design of collaborative visualization.

Collaborative analysis is a necessary and promising direction to handle the explosion of data volume and visualization challenges [58]. Such techniques are often required for application fields where task complexities can easily overrun computing power and algorithm intelligence. Especially in security visualization, efficient collaboration mechanisms are crucial to provide time efficient solutions that can process a large amount of data in real time. While visualization techniques have been explored for assisting individual analysts with detection tasks, the problem of collaborative analysis for such applications is open. This chapter is to provide useful information on the designs of such collaborative analysis systems through a formal user study.

Our motivation comes from the fact that security teams cost too much time for protecting networking environments traditionally. In order to improve the efficiency of security teams in network protection, as a result, more security teams are now deploying visualization systems that allow experts to monitor the security state of a network. Collaborative visualization can provide a practical solution to overcome the ineffectiveness of automatic detection algorithms, limits of computing resources, and complexity of advanced malicious attacks. Especially for intrusion detection, where new or unknown attacks are often introduced, having a group of experts analyzing data effectively in real time is crucial to provide accurate and time critical results.

In this chapter, we design and perform a formal user study to evaluate collaboration of participants in different aspects of collaborative visualization on intrusion detection. The common collaborative detection aspects include task prioritization, task selection, task coordination, and communication. Our study compares these aspects under two important collaboration environments: distributed and co-located settings, which are both possible solutions for intrusion detection tasks. To provide general information on collaborative intrusion detection, we design our detection tasks solely based on the identification of visual patterns from simulation datasets. User performance and detection accuracy are analyzed and summarized.

The main contribution of this chapter is that our evaluation study explores and summarizes collaboration strategies for network security applications. Our results can provide guidelines for future design and development of collaborative security visualization systems. The coordination and communication strategies can also be used for general collaborative visualization design.

7.2 User Study for Collaboration Visualization

To design collaborative intrusion detection methods we need to understand how these methods are used by teams in different environments. In particular, we need a better understanding of how team members coordinate their activities for intrusion detection tasks. Our study is designed to provide the comparisons of user collaboration in distributed and co-located environments respectively. In this exploration, we are guided by the common collaborative detection aspects from coordination and communication in [71], which include task prioritization, task division, conflict solving, decision making and knowledge sharing.

We start from describing distributed and co-located collaboration environments briefly. Then we introduce the detection goal and visualization methods applied in the user study. The details of our user study design are presented at last.

7.2.1 Distributed and Co-located Collaboration Environments

The distributed setting is often used for applications which allows users from different locations to access data remotely and shares data analysis results through an web-based interface. For example, security data can be collected and stored at one or multiple servers. Users from different locations may access all or portions of the data according to the collaboration design. Generally users share their results of data analysis by exchanging snapshots from visualization systems or summaries directly.

Different from distributed setting, the co-located setting is used for collaboration environments such as multi-panel or tabletop displays. Users may have a common visualization space as well as their own work spaces. For example, several analysts can gather around a tabletop display and analyze a complex networking attack. Communications for both data analysis and conclusions are made face to face directly.

Both distributed and co-located collaboration environments can be adopted for intrusion detection. Comparing these two collaboration environments, distributed teams involve less social interaction and focus more on analysis tasks. However, one major problem is to maintain trust among team members, since it is hard for them to keep track of the other members' work status. Conversely, co-located teams are more socially oriented and thereby maintaining high cohesion among team members [136]. On the other hand, co-located team members may have difficulty to express their conflict ideas.

7.2.2 Intrusion Detection with Visualizations

This user study concentrates on detecting Sybil Attacks collaboratively under the distributed and co-located environments. Three detection strategies and visualization have been introduced in Chapter 5. As demonstrated in Figure 39, a matrix visualization of normal network topology generally has an appearance of random pattern as shown on the left image of Figure 39. While the middle and right patterns, generated by reordering the node sequences, are indications of potential Sybil attacks. The white nodes located on the left bottom corners are suspicious in such patterns. In our study, as long as one of the visualization image demonstrates the suspicious patterns, participants can identify Sybil attacks in a respective time period.

Additionally, 2D histogram visualization has been used as shown on Figure 40,

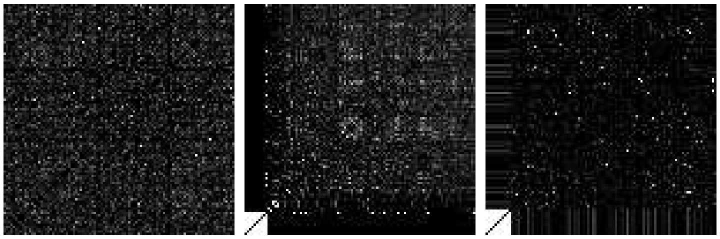


Figure 39: Left: general topology matrix does not reveal suspicious patterns. Middle and Right: a suitably reordered topology matrix with a certain time range can reveal traces to identify malicious nodes.

which can reflect data properties along the time axis based on attack features. The time histogram is a grey scale image in the space of node index and time step. For every node at each time step, its "significance" value is measured corresponding to the attack features, and a histogram is generated by linearly mapping all the significance values onto grey colors. The significance values are calculated by the following procedure. At each time step, the nodes are grouped together if they shared the same set of neighbors. With this method, each node belongs to only one group at each time step. The significance value of a node is set as the size of its group divided by the largest group size at this time step. Larger significance values are painted with brighter colors. The time histogram is generated by traversing all the time steps.

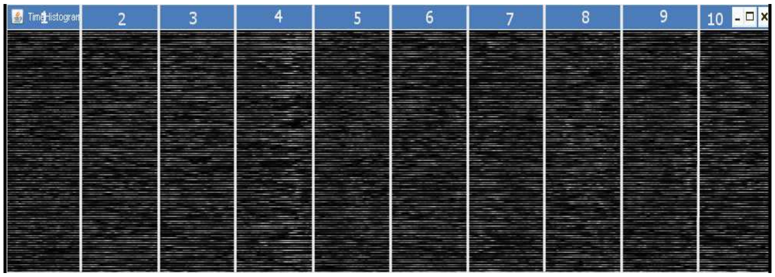


Figure 40: In the time histogram, bright lines indicate suspicious degree of durations. The fourth duration has the highest suspicious degree because of its brightest lines.

This time histogram assists participants to identify attack durations. Since fake identities of Sybil attacks usually appear and move in groups during a certain time period, they share a large portion of neighbors for multiple time steps. The time histogram collects this grouping information and produces obvious patterns along the time axis.

In conclusion, the matrix and 2D histogram visualizations have been used in different steps in our user study to achieve different goals.

7.2.3 Design of User Study

In order to gain insight into what features are necessary for effective collaborative security visualization systems, a user study has been conducted to examine collaborative behaviors in distributed and co-located settings. To isolate effective collaboration designs from specific detection methods, all data interaction capabilities have been removed and only tasks related to visual exploration of image patterns have been used in the user study.

During the study process, participants were asked to perform collaborative intrusion detection through finding certain visual patterns under the settings of distributed and co-located collaboration respectively. Our study was divided to several different steps including task initialization, division, classification, modification, confirmation and decision making. In the study, participants were observed by investigators to record their collaboration reactions. Both the observation and results of participants were used to measure user performance and accuracy.

Matching effective communication strategies with successful participant performance yields insights into what collaborative features should be included in a visualization and at what point in the collaborative analysis process they should be made available to the participant. The following describes the details of subjects, materials, and procedure of our user study respectively.

7.2.3.1 Participants

The participants in the study were 10 volunteers, 2 females and 8 males. They were graduate students in the major of computer science. About half of the participants had visualization or computer graphics backgrounds. The ages of participants ranged from 20 to 30. The ten participants formed five collaboration paired teams. For all the five teams, the 2 participants in each pair had known each other before the study. They had previously co-working experience. It is consistent with real-life scenarios that collaborators are often acquaintances.

In our study, each pair of users were required to do collaborative detection under one of distributed and co-located environments firstly, then they changed their collaboration environment to detect attacks with a different dataset.

7.2.3.2 Materials and Interface

The materials were images generated with a set of network simulation datasets. Two datasets were used to switch for different collaboration environments. There were one hundred nodes and ten thousand time steps in each dataset. In the first dataset, three Sybils existed at two durations of 3000-4000 and 7000-8000; in the second dataset, five Sybils existed at two durations of 2000-4000 and 6000-8000. To simulate the difficulty in selecting proper tasks, we divided each dataset into ten equal parts. For each part, we generated a matrix visualization image group including one general topology matrix and three reordered topology matrices by the three detection algorithms. A task was defined as the detection of suspicious patterns from an image group.

As shown in Figure 41, some matrix visualization images contained random noises, and the others contained visual patterns at different degrees. We defined the suspicious degree of an image as the significance of patterns from noises in perception. For example, the suspicious degrees of the images from the first to the third row in Figure 41 were from low to high, as the third row contained obvious square patterns which were considered to be malicious.

We had carefully selected the ten tasks for each simulation set, so that they included all the three cases of no attacks, subtle attacks, and obvious attacks. Since part of an effective security visualization was able to confirm a "safe" state in the network, it was crucial to include tasks without attacks.

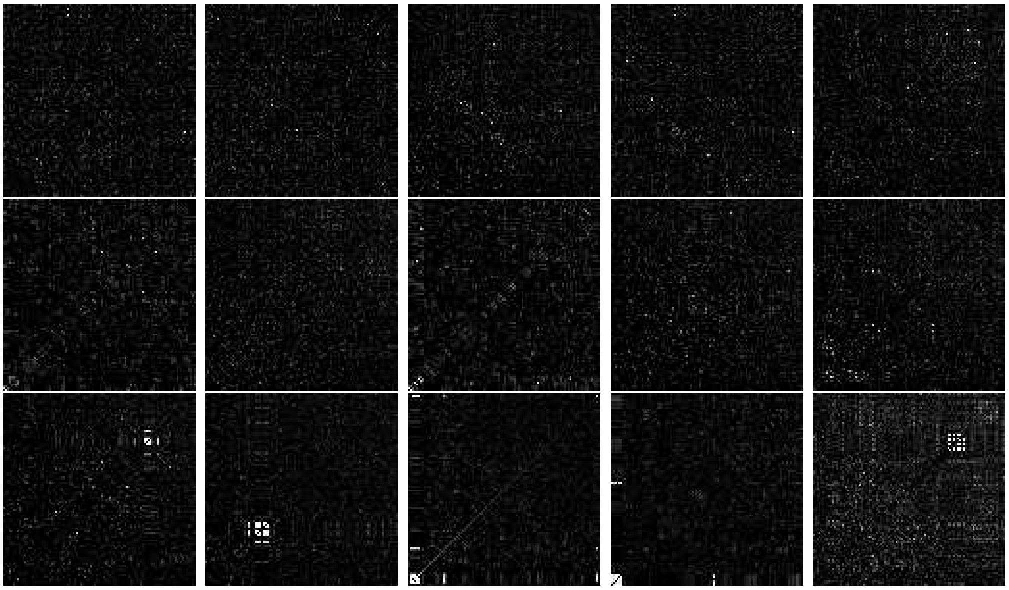
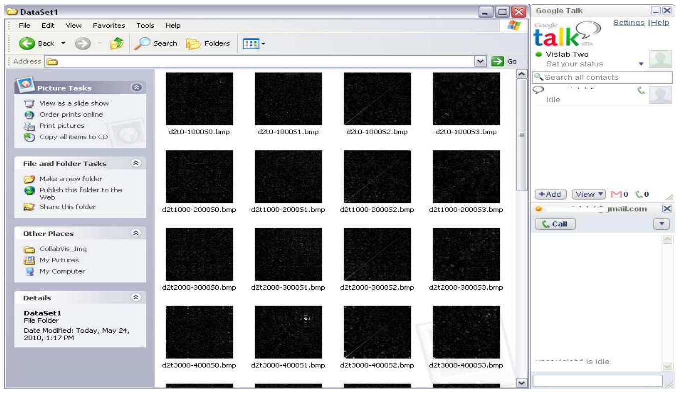


Figure 41: Examples of images used in the study. The patterns from the first to the third row were ranging from random to obvious. Correspondingly, their suspicious degrees were from low to high.



DataSet1
File Folder
Date Modified: Today, May 24, 2010, 1:17 PM

Figure 42: Interface of User study

To isolate the effects of coordination from detection strategies, we kept our study interface simple. As shown in Figure 42, the materials were arranged in the Windows Explorer on the left. Each row showed four images belonging to the same task. Users could easily view images in rows and enlarge a specific one during the analysis process. This simulated a small-multiples view, which allowed users to rapidly identify patterns that indicated possible attacks. On the right of the screen, the Google Chat windows were located for communication, especially for distributed teams.

7.2.3.3 Procedure

Our study was composed of two trials under distributed and co-located collaboration environments. The orders of the two trials were altered for the five collaboration teams. Two groups started with the distributed setting and the other three with the co-located setting. Similarly, the orders of the use of dataset were altered. This design balanced potential differences caused by the trail order.

Each computer was set up with a Google Chat session, small-multiples view, and Google Spreadsheet for inputting analysis results. In the co-located setting, participants shared a computer. In the distributed setting, participants were placed on two separate computers and they were instructed to communicate via the Google chat. For each finding, participants followed a specific text format to report results, like "Dataset: 1, Task: 2, Classification: maybe attack, Notes: three suspicious nodes in top right corner of visualization 2". The fixed text format allowed us to collect data statistics automatically.

After explaining the goal of the study, we held a training session for each pair of participants. During the training session, we demonstrated the procedure of data analysis with an example dataset. All components including visual pattern exploration, communication chat session and spreadsheet session were explained in detail to ensure that all participants were able to complete tasks successfully.

Except the differences between distributed and co-located collaboration settings, the procedures for their trials were the same. The participants were instructed to detect all the suspicious patterns among the provided materials through six steps, which corresponded to different stages of collaborative analysis.

Step 1 - Task initialization: Participants started by identifying the suspicious degrees for ten tasks based on the time histograms (Figure 40). A pair of participants worked together to assign each task a reward value, which was designed to represent the suspicious degree of each task. The reward values were scaled from one to ten, with ten being most suspicious degree. These initial reward values were recorded in a spreadsheet.

The results of task initialization provided participants an estimation of the time cost for each task. The team members could use this information to negotiate a suitable task division in the next step.

Step 2 - Task division: The second step was to divide tasks between the two partners. Instead of assigning tasks to participants, our study allowed each team to divided the tasks actively through negotiation and their initialization of tasks. The behaviors of the participants were observed closely during this step. We believed that the design of this step could come up with different effective collaborative analysis strategies.

Step 3 - Detection via task classification: After dividing the tasks, participants started to analyze their tasks by studying the corresponding images in the small-multiples panel and classifying the tasks into four categories. These four categories were defined including definitely no attack (DNA), maybe no attack (MNA), maybe attack (MA), and definitely attack (DA). The highest suspicious degree among the four images belonging to the same task was used to assign the task category and was recorded into spreadsheet. These categories were used not only to measure analysis accuracy, but also for communication (especially in the distributed setting) since the spreadsheet was updated at all locations in real time.

Step 4 - Modification of task reward value: After the detection step, each participant could modify the reward values of his or her tasks according to the detection results. The modified task rewards were marked. This modification emphasized the uncertain and complex tasks with dramastically increasing or decreasing of reward values. It also helped the partners to effectively re-analyze these tasks, which was the next step.

Step 5 - Confirmation: In order to observe how team members solve conflicts in collaboration, we designed this step for participants to switch tasks and check the detection classifications of their partners. We compared how participants challenged the decisions of their partners under two different environments. This step was important for the two partners to achieve consensus results.

Step 6 - Decision Making: Conflict was a common issue in collaboration work. After the step of confirmation, it was unavoidable that conflict results of some tasks existed between team members. In order to solve the conflicts, the paired teams discussed to make a final decision for each task on its suspicious category. We observed the behaviors of paired teams to solve conflict opinions and the way they made decisions under different conditions. We studied the factors that influenced the decision making for collaboration detection.

7.3 Results and Analysis

In this section, we present the data analysis of the user study results and discuss our findings. For clarity, we assigned a unique label in these results for each group (G)-environment (C)-dataset (D). For example, G1-C1-D1 indicated the first collaboration team performed under distributed environment to analyze the first dataset. G3-C2-D2 indicated the third collaboration team performed under the co-located setting to analyze the second dataset. Based on the results of the user study, we analyzed the coordination and user communication aspects under distributed and co-located environments.

7.3.1 Coordination

As described in [82], coordination is the attempt by multiple entities to act in concert in order to achieve a common goal by carrying out a script/plan they all understand. Thus building up such a good script for team members is the key point for the whole collaboration process. Coordination should be defined by the combination of tasks, tools, and communication [98]. In this section, we discuss the procedures and the value of coordination including the updating of reward values, the task division, the task confirmation, and the decision making.

The updating of reward values: As we introduced, paired participants made an initial assumption together for each task based on time histogram visualization. After detection with matrix visualization, the reward values for suspicious degree were modified if necessary. We observed that almost all participants started their assumptions for the tasks with brighter lines in time histogram. For these higher suspicious tasks, they could make quick assumptions. For example, the durations of 3000-4000 (task 4) and 7000-8000 (task 8) of the first dataset had been assigned high reward values both in distributed and co-located settings as shown in Figure 43. In intrusion detection, these suspicious durations should be drawn more attentions than the other durations. Therefore, the time histogram visualization, as a general view for all durations of network dataset, was demonstrated to be helpful for improving the efficiency of collaboration detection.

From Figure 43, we could find that the curves of average reward values were very similar for the first four tasks under two environments, but quite different for tasks 6, 7

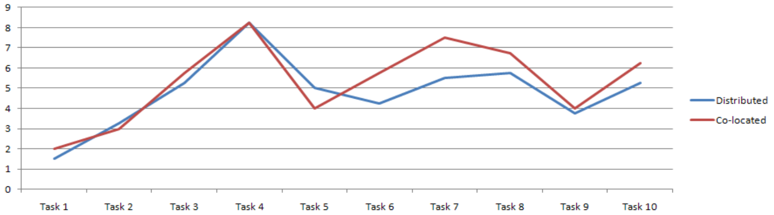


Figure 43: Task initialization for average reward values assumption of data 1.

and 8. The patterns of these tasks were not obvious in time histogram. The co-located setting tended to lead to higher reward values for these tasks than the distributed setting. Since detection and task analysis had not been introduced at this step, we analyzed the reasons for causing the differences through observing the communication between team participants. We found that the co-located team participants had much more communication to share their opinions directly. They had stronger confidence on their choices of the reward values. Also, the convenient communication brought more trust for teams. Conversely, the distributed teams were more conservative as fewer communication through Google chat session. Their communication was simple without much knowledge sharing. In summary, co-located teams made audacious assumptions while distributed teams made conservative assumptions for these unclear tasks.

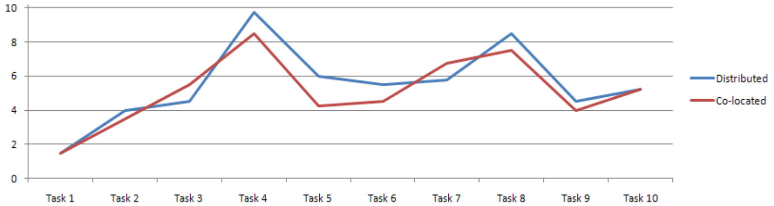


Figure 44: The modification of average reward values after task detection for data 1.

Interesting thing happened after participants finishing the task detection with matrix visualization. Figure 44 showed the modification of average reward values for data 1. The distributed teams updated higher reward values for the suspicious tasks than the co-located teams, which was different from Figure 43. This indicated that distributed setting leaded to more accurate detection than the co-located setting. We observed the Google chat history of distributed teams and found that the participants in distributed teams stopped their communication during the detection step. They worked independently and focused on solving the tasks by themselves completely. Differently, the co-located teams kept communication in every step of user study. In summary, the focus on task analysis influenced the performances of participants for distributed and co-located teams.

The task division: In our study, we allowed the paired participants to divide the tasks actively by themselves. We concluded three strategies of task division for paired collaboration teams with different backgrounds and environments.

The first strategy divided tasks equally, such as the G1-C1-D1 in Figure 45 assigned five tasks 2, 3, 5, 8 and 9 to one participant and the rest to the other. This team assigned the highest reward values to tasks 3 and 7 and separated them to the team participants fairly. Similarly, G1-C2-D2, G2-C2-D1, G4-C1-D1, G5-C1-D2, and G5-C2-D1 selected tasks equally.

The second strategy of task division was to divide tasks to balance workload. For example, in Figure 45, the tasks of G4-C1-D1 in green included one highest reward value task with other five lower reward value tasks. The suspicious degrees of red tasks were in the middle levels. This strategy gave the first participant more time to focus on the most suspicious tasks and balanced the workload. Similar case included G3-C2-D1.

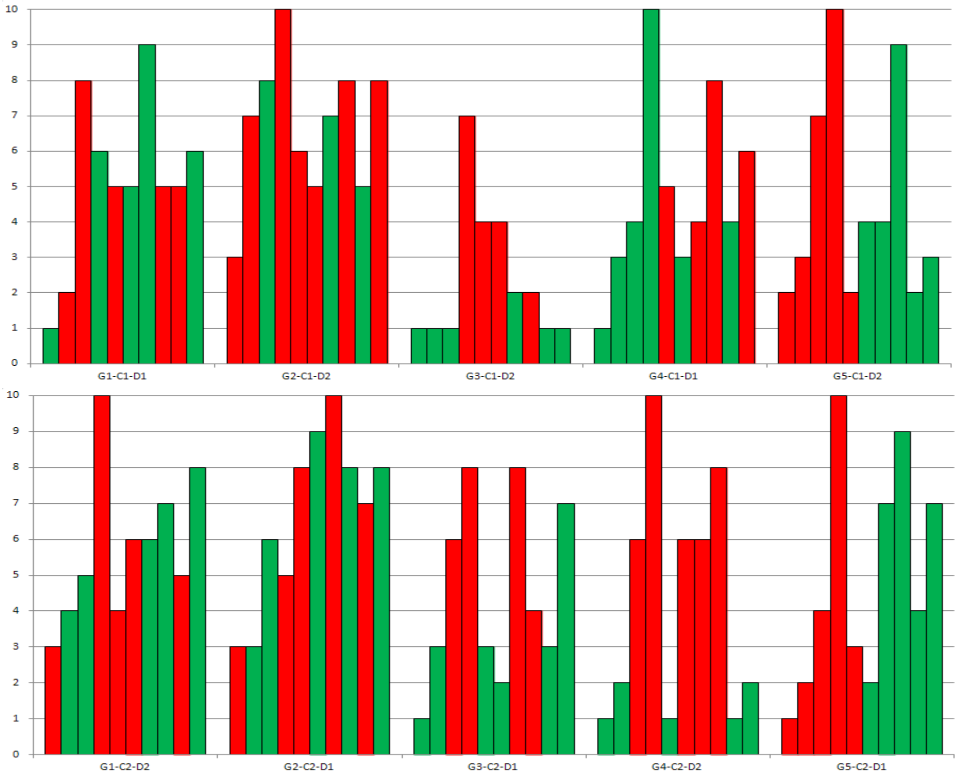


Figure 45: Task division with reward values. The upper figure is the task division under distributed settings; the bottom figure is the task division for co-located teams. There are five paired teams under each environments. In each team, the reward values of ten tasks are plotted. Red and green colors are used to indicate the tasks are divided to different participants for each team.

The third strategy of tasks division was based on background and interests of participants. In G2-C1-D2 (Figure 45), one participant took seven tasks (red) including the most suspicious one. This participant had visualization background and he was interested in our collaboration detection visualization. In G3-C1-D2 and G4-C2-D2, the tasks with high reward values were all assigned to one partner, as they had past experience on network security analysis. This strategy emphasized the results of participants with related background.

The task confirmation: We further studied how participants communicated with their partners when they had conflict opinions. We analyzed the modified results of all tasks and the talk or chat histories during the user study.

Table 3: Results of task classification and re-classification under the distributed environment. The initial classification results are in front of the re-classification results. The suspicious tasks of D1 are task 4 and task 8; the suspicious tasks of D2 are task 3, 4, 7 and 8. The classification results of DA and MA for suspicious tasks were considered as successful detection.

Tasks	G1-C1-D1	G2-C1-D2	G3-C1-D2	G4-C1-D1	G5-C1-D2
task 1	DNA/MNA	MNA/MNA	MNA/MNA	DNA/DNA	MNA/DNA
task 2	MA/DNA	MA/MA	MA/MA	DNA/MNA	MNA/MNA
task 3	MNA/MNA	MA/MA	DA/MA	DNA/MNA	DA/MA
task 4	DA/DA	DA/DA	DA/DA	DA/DA	DA/DA
task 5	DA/MA	MA/MNA	MNA/MNA	MA/MA	DNA/DNA
task 6	MNA/DNA	MA/MA	MA/MA	MA/MA	MNA/MNA
task 7	MA/MA	DA/MA	MA/MA	MNA/MA	MNA/MNA
task 8	DA/DA	DA/DA	MA/MA	DA/DA	DA/DA
task 9	MNA/MA	MA/MA	MA/MNA	MA/MA	DNA/DNA
task 10	MA/MA	MA/MA	MNA/MA	MA/MA	MNA/DNA

Table 4: Results of task classification and re-classification under the co-located environment. The initial classification results are in front of the re-classification results.

Tasks	G1-C2-D2	G2-C2-D1	G3-C2-D1	G4-C2-D2	G5-C2-D1
task 1	DN/DN	DNA/MNA	MNA/DNA	DNA/DNA	DNA/DNA
task 2	MA/MA	MA/MNA	MNA/MNA	MA/MA	DNA/DNA
task 3	MA/MA	DA/MA	MNA/MNA	MA/MA	MA/MNA
task 4	DA/DA	MNA/DA	MA/DA	DA/DA	MA/DA
task 5	MNA/MNA	DA/MA	MA/MNA	DNA/MNA	MNA/DNA
task 6	MNA/MA	MNA/MA	DNA/MNA	MNA/MNA	DNA/DNA
task 7	MNA/MNA	MA/MA	MNA/MA	MA/MA	MA/MA
task 8	DA/DA	MA/DA	MNA/DA	DA/DA	DA/DA
task 9	MA/MA	MA/MA	DNA/MA	DNA/MNA	MNA/MNA
task 10	MA/MA	DA/DA	MA/MA	DNA/MA	MA/MA

Tables 3 and 4 showed the results of task classification and re-classification under the distributed and co-located environments. For these results, we considered tasks with classification of DA and MA as suspicious, while tasks with classification of DNA and MNA as unsuspicious. For distributed teams, we found that different classification results appeared frequently for the tasks adjacent to the suspicious tasks and the classifications for the suspicious tasks were all correct. We observed that distributed teams showed their doubts to their partners directly by modifying the classification results mainly based on their own judgement. In contrast, co-located teams discussed the results whenever there were different opinions. Although this step was not for the final decision, the two partners all tried to reach a consensus. Co-located teams demonstrated a better trust between the team partners. Also, we observed that the co-located team did not challenge their partners as frequently as distributed teams.

Additionally, allowing participants to confirm their partners' work increased the accuracy of the detection results. For example, in Table 3, group G1-C1-D1 enhanced their classification of task 2 from MA to DNA. In Table 4, group G5-C2-D1 modified the task 3 results from MA to MNA. Especially for co-located teams, the accuracy of suspicious tasks improved significantly. For example, in Table 4, group G2-C2-D1 explored the suspicious task 4 from MNA to DA; group G3-C2-D1 identified the suspicious task 8 from MNA to DA.

The decision making: Understanding how participants reached their decisions was important and helpful to improve the efficiency of collaboration. There were a number of factors influencing decision making, such as participants differences, trusts, sharing information, and cognitive biases. We analyzed the decision making process and summarized three decision making styles in our study.

The first style was the decision made by a single participant. This style appeared in the teams of participants with different background. For example, in one team, one participant had research experience related to network security and the other participant had not. The participant with network security background hold a leading position in task analysis. Under this style, participants could make decisions easily and solve conflicts quickly.

The second style was simply averaging the results of two partners. The reward value was the main factor for final dicisions. The participants calculated the average reward values of each task in the detection and confirmation steps. The average reward value in this way was a compromise result for both participants.

The third style was to make decisions through negotiation. Negotiation for decisions required more trust, sharing information, and effective communication in teams than the previous two styles. Therefore, most co-located teams made decisions in this style because of their convenient communication environment. This style took both participant opinions into account and the decisions were fully supported by the teams.

The three decision making styles had limitations respectively. Decision by a single participant did not take the advantage of the whole group; decision by averaging might ended with results that no one in the team fully agreed; decision by negotiation costed more time as it involved of more communication and knowledge sharing. In conclusion, different style of decision making should be designed for different conditions and requirements.

7.3.2 Communication

For better understanding what information the collaboration teams exchanged and the reasons for the communication, we analyzed the teams in detail under both distributed and co-located collaboration environments.

Communication occurrences: In our study, communication occurred frequently in the steps of task initialization, task division and decision making. For co-located environments, communication also happened in the step of task confirmation.

During task initialization, participants observed time histogram and compared the brightness of time histogram for each task. They discussed the suspicious time durations and assigned the reward values. These communication helped participants understand each other's security background and the differences of tasks.

Based on the task initialization results, participants communicated to divide these tasks. They shared their past work experience and their interests for intrusion detection tasks. Usually, they tried to balance the workload for each other. It was possible that some experienced participants received more or heavier tasks, which maximized their abilities in detection.

Decision making needed communication for both distributed and co-located environments. Participants discussed their findings and reasons for the conflicted tasks and verified them. The participants zoomed the matrix visualization for details and shared their confidence and evidence to their partner, and tried to convince their partner. Through their performance and their experience, they made decisions by one of the three styles described earlier.

Communication was more engaged in the step of task confirmation in co-located environment than distributed environment. Co-located participants expressed their doubts with reasons to their partners and tried to modify conclusions with their partners' agreement.

The value of communication: Communication was an important issue in our collaboration detection and it occurred in almost all tasks. The values of communication between team participants included:

1. building trust between participants
2. focusing on suspicious tasks and improve efficiency
3. maximizing participants' ability and improve accuracy
4. solving conflicts with evidence
5. reducing the cost of detection

Communication was more common in co-located settings and it needed to be improved for distributed teams. In distributed teams, though the Google spreadsheet was provided for team members to keep work awareness, the information sharing especially the reasons of their findings was still weak.

7.4 Discussion

In this section, we discuss the general performance of collaboration teams under distributed and co-located environments. We also talk about two human factors, trust and sharing for decision making, which should be enhanced in our design.

Table 5: Average response time for each step.

Steps	Distributed(min)	Co-located(min)
Step 1	4:51	3:16
Step 2	0:39	0:41
Step 3	3:10	5:10
Step 4	1:58	1:30
Step 5	3:13	4:16
Step 6	10:19	9:55

From Tables 3 and 4, the teams detected all suspicious tasks except the first group did not find the malicious patterns for task 7 under co-located setting. Generally, their performances on the accuracy were almost the same. However, we found that the time they spent on each task was different. We recorded the response time of collaboration teams for each step and calculated the average response time shown in Table 5. From the results, we found that distributed teams performed faster in steps 3 and 5 while spent more time on the other tasks than co-located teams. Since steps 3 and 5 were task detection and confirmation, both of them needed the participants to focus on tasks. Distributed teams was in task-related working style as they were not distracted by the other things, such as the partners activities. While the steps 1, 2, 4, and 6 needed teams had a good communication environment, which showed the benefits of co-located settings. In conclusion, communication was the "lubricant" to make collaboration better. On the other side, too much communication could distract the focus of teams.

The final step in our study was to make decisions for tasks. Usually, conflicts were inevitable for group decision making because of the task uncertainty and individual differences. Each participant used his domain knowledge to make a decision. The decision might be disagreed by the participant's partner. In such a situation, participants had to collaborate and exchange information to reach a consensus. Thus, human-related issues such as trust and sharing needed to be brought into the procedure of decision making.

One kind of human trust was built by familiarity or similarity belief [73]. In our study, the paired participants had such direct trust as they had previous co-working experience. Another kind of trust is built dynamically during the detection process.

We observed the division strategies usually changed in our study. For example, group 2 divided the tasks equally in the beginning under co-located setting. But after they finished the first study, one participant were assigned seven tasks in the second study under distributed setting. We found this participant showed a big interest and took a leading position in the first study, and his partner agreed with him most of the time. Thus, his partner trusted him for his previous performance and agreed to assign more tasks to him in the second study. Similar cases happened in groups 3 and 4. They initially used the division strategies of workload balance, then they switched their division strategies based on their increased trusts between team participants.

Sharing information was another factor to solve conflicts. Sharing information did not only allow participants to exchange their findings, but also the reasons behind the findings. In co-located environment, participants could point out their findings in matrix visualization images and talk about their findings with conclusion and suggestion to their partners. But we observed that for distributed teams, they only shared their results and hardly showed their reasons. Thus, methods to help distributed participants sharing their findings with reasons were needed to improve the performance of distributed teams to make decisions collaboratively.

7.5 Conclusion and Future Work

Collaborative analysis is one practical solution for important and time-critical intrusion detection tasks. This chapter presents a formal user study to evaluate several aspects of collaborative detection design, including coordination and communication, through exploration tasks with visual patterns. We conclude several strategies on task division and decision making in coordination. We observe and analyze the communication occurrences to summary the values of communication. Our study compares two common collaboration environments, which can both be used for intrusion detection systems. Our results and data analysis provide useful insights to design and evaluate collaborative intrusion detection methods.

In the future, we plan to use our summarized strategies of communication and coordination to improve design of the other security systems. This study can also be extended to other collaborative visualization applications.

CHAPTER 8: CONCLUSIONS

This dissertation provides new visualization methods for exploring network topology efficiently through two aspects: spectrum-based network analysis and collaborative visualization technique.

For spectrum-based network analysis, we explore unsigned and signed network topology structure by utilizing the features of node distribution and coordinates in the spectral space. Our methods can assist users to reveal the global topology structure, the importance of individual nodes and edges to each network community, and the relationships of friends or foes among the members for unsigned and signed networks. Our approaches include an automatic and non-iterative network layout algorithm and several interactive visual analytics tools for analyzing complex network topologies. Several additional spectrum-based features, such as a consistent framework of edge and node randomness measurements, are integrated in our methods. Comparing to the other approaches, our methods are demonstrated to be efficient and effective for complex network analysis.

For collaborative visualization technique, we incorporate results from models of human behavior, teamwork theory, and interface design to propose collaborative visualization framework to improve the efficiency for network visualization analysis by multiple users. In our design, we use a network security application which analyzes the features of network topology under various Sybil attacks. We propose several guidelines and heuristics to enable multiple users collaborate effectively to detect suspicious attacks, sharing information, and communication. Also, we design a user study to evaluate how users collaborate to tackle the problems under different collaboration environments, which are both available for real network visualization applications. We conclude several coordination strategies and summarize the values of communication for collaborative visualization.

In the dissertation, our approaches have been demonstrated to be efficient with both synthetic and real-life networks. The computation complexity and scalability of our network analysis approaches prove the potential of our approaches to handle large-scale networks. Our approaches can also provide helpful information for future design and development of collaborative visualization systems. We believe the analysis of network topology is a field with much potential to explore in the future.

REFERENCES

[1] Abuaitah, G. R., and Wang, B. Secvizer: A security visualization tool for qualnet- generated traffic traces. VisSec poster, 2009.

[2] Albrecht, M., Kerren, A., Klein, K., Kohlbacher, O., Mutzel, P., Paul, W., Schreiber, F., and Wybrow, M. On open problems in biological network visualization. In *Graph Drawing* (2010), Springer, pp. 256–267.

[3] Ancona, M., Cazzola, W., Drago, S., and Quercini, G. Visualizing and managing network topologies via rectangular dualization. In *IEEE Symposium on Computers and Communications* (2006).

[4] Andrews, K., and Heidegger, H. Information slices: Visualising and exploring large hierarchies using cascading, semi-circular discs. In *Proc of IEEE Infovis 98 late breaking Hot Topics* (1998), pp. 9–11.

[5] Archambault, D., Munzner, T., and Auber, D. Topolayout: Multilevel graph layout by topological features. *Visualization and Computer Graphics, IEEE Transactions on* (2007), 305 –317.

[6] Arvo, J. Techniques for interactive graph drawing. In *Revised Papers from the 10th International Symposium on Graph Drawing* (2002), Springer-Verlag, p. 380.

[7] Au, S., Leckie, C., Parhar, A., and Wong, G. Efficient visualization of large routing topologies. *International Journal of Network Management 14*, 2 (2004), 105–118.

[8] Auber, D., Chiricota, Y., Jourdan, F., and Melançon, G. Multiscale visualization of small world networks. In *Proceedings of the Ninth annual IEEE conference on Information visualization* (2003), INFOVIS'03, pp. 75–81.

[9] Bachmaier, C., Brandenburg, F., Forster, M., Holleis, P., and Raitner, M. Gravisto: Graph visualization toolkit. In *Graph Drawing* (2005), Springer, pp. 502–503.

[10] Ball, R., Fink, G., Rathi, A., Shah, S., and North, C. Home-centric visualization of network traffic for security administration. In *Proc. of ACM VizSEC/DMSEC* (2004).

[11] Bannister, M. J., Eppstein, D., Goodrich, M. T., and Trott, L. Force-directed graph drawing using social gravity and scaling. *arXiv preprint arXiv:1209.0748* (2012).

[12] Bogdanov, P., Larusso, N., and Singh, A. Towards community discovery in signed collaborative interaction networks. In *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on* (2010), pp. 288 –295.

[13] Boitmanis, K., Brandes, U., and Pich, C. Visualizing internet evolution on the autonomous systems level. In *Graph Drawing* (2008), Springer, pp. 365–376.

[14] Brandes, U., Fleischer, D., and Lerner, J. Summarizing dynamic bipolar conflict structures. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1486–1499.

[15] Brandes, U., and Lerner, J. Visual analysis of controversy in user-generated encyclopedias. In *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology* (2007), VAST '07, pp. 179–186.

[16] Brandes, U., and Pich, C. Eigensolver methods for progressive multidimensional scaling of large data. In *Graph Drawing* (2007), Springer, pp. 42–53.

[17] Bresciani, S., and Eppler, M. J. The benefits of synchronous collaborative information visualization: Evidence from an experimental evaluation. *Information Transaction on visualization and Computer Graphics* (2009), 1073–1080.

[18] Carriere, J., and Kazman, R. Research report: Interacting with huge hierarchies: beyond cone trees. *Information Visualization, IEEE Symposium on 0* (1995), 74–81.

[19] Cartwright, D., and Harary, F. Structural balance: a generalization of heider's theory. *Psychological Review 63*, 5 (1956), 277–93.

[20] Castro, M., Druschel, P., Ganesh, A., Rowstron, A., and Wallach, D. S. Secure routing for structured peer-to-peer overlay networks. In *USENIX OSDI* (2002), pp. 299–314.

[21] Chan, Y.-H., Keeton, K., and Ma, K.-L. Interactive visual analysis of hierarchical enterprise data. *E-Commerce Technology, IEEE International Conference on 0* (2010), 180–187.

[22] Chernobelskiy, R., Cunningham, K., Goodrich, M., Kobourov, S., and Trott, L. Force-directed lombardi-style graph drawing. In *Graph Drawing* (2012), Springer, pp. 320–331.

[23] Chung, F. *Spectral Graph Theory.* American Mathematical Society, 1997.

[24] Clark, H., and Brennan, S. Grounding in communication. *Perspectives on socially shared cognition* (1991), 127–149.

[25] Clark, H. H. Pointing and placing. In S.Kita(Ed),Pointing, Where language, culture, and cognition meet, 2003.

[26] Convertino, G., Ganoe, C., Schafer, W., Yost, B., and Carroll, J. A multiple view approach to support common ground in distributed and synchronous geo-collaboration. In *Coordinated and Multiple Views in Exploratory Visualization, 2005. (CMV 2005). Proceedings. Third International Conference on* (july 2005), pp. 121–132.

[27] COW. Correlates of war datasets. http://www.correlatesofwar.org/.

[28] Cui, W. A survey on graph visualization. Hong Kong University of Science and Technology.

[29] Cui, W., Zhou, H., Qu, H., Wong, P. C., and Li, X. Geometry-based edge clustering for graph visualization. *IEEE Transactions on Visualization and Computer Graphics 14* (November 2008), 1277–1284.

[30] Cummings, J. N. Work groups, structural diversity, and knowledge sharing in a global organization. *Management Science* (2004), 352–364.

[31] Davis, J. Clustering and structural balance in graphs. *Human Relations* (1967).

[32] Di Giacomo, E., Didimo, W., Grilli, L., and Liotta, G. Whatsonweb: Using graph drawing to search the web. In *Graph Drawing* (2006), Springer, pp. 480–491.

[33] DiCaterino, A., Larsen, K., Tang, M.-H., and Wang, W.-L., 1997. An Introduction to Workflow Management Systems Models for Action Project:Developing Practical Approaches to Electronic RecordsManagement and Preservation.

[34] Didimo, W., Liotta, G., and Romeo, S. Topology-driven force-directed algorithms. In *Graph Drawing* (2011), Springer, pp. 165–176.

[35] Dömel, P. Webmap - a graphical hypertext navigation tool. In *Proceedings of the Second International World Wide Web Conference* (1994), vol. 28, pp. 85–97.

[36] Douceur, J. R. The sybil attack. In *the First International Workshop on Peer-to-Peer Systems* (2002), pp. 251–260.

[37] Dumas, M., McGuffin, M., Robert, J.-M., and Willig, M.-C. Optimizing a radial layout of bipartite graphs for a tool visualizing security alerts. In *Graph Drawing* (2012), Springer, pp. 203–214.

[38] Dyck, B., Joevenazzo, J., Nickle, E., Wilsdon, J., and Wismath, S. Gluskap: Visualization and manipulation of graph drawings in 3-dimensions. In *Graph Drawing* (2004), Springer, pp. 496–497.

[39] Eades, P. A heuristic for graph drawing. *Congressus Numerantium 42* (1984), 149–160.

[40] Eades, P., and Huang, M. L. Navigating clustered graphs using force-directed methods. *Journal of Graph Algorithms and Applications 4* (2000), 157–181.

[41] E.Brennan, S., Mueller, K., Zelinsky, G., Ramakrishnan, I., S.Warren, D., and Kaufman, A. Toward a multi-analyst, collaborative framework for visual analytics. *Proceedings of IEEE VAST* (2006).

[42] Egidi, M., and Marengo, L. Division of labor and social coordination modes: A simple simulation model. *Simulating Societies* (July 1993).

[43] Estrada, E., and Rodruez-Velquez, J. A. Subgraph centrality in complex networks. *Physical Review E 71(056103)* (2005).

[44] Finkel, B., and Tamassia, R. Curvilinear graph drawing using the force-directed method. In *Graph Drawing* (2005), Springer, pp. 448–453.

[45] Foresti, S., Agutter, J., Livnat, Y., Moon, S., and Erbacher, R. Visual correlation of network alerts. *Computer Graphics and Applications, IEEE 26*, 2 (2006), 48–59.

[46] Forrester, D., Kobourov, S., Navabi, A., Wampler, K., and Yee, G. graphael: A system for generalized force-directed layouts. In *Graph drawing* (2005), Springer, pp. 454–464.

[47] Fruchterman, T. M. J., and Reingold, E. M. Graph drawing by force-directed placement. *Softw. Pract. Exper. 21* (November 1991), 1129–1164.

[48] Fu, Y., Chase, J., Chun, B., Schwab, S., and Vahdat, A. Sharp: an architecture for secure resource peering. In *Proc. of the nineteenth ACM symposium on Operating systems principles* (2003), pp. 133–148.

[49] Fussell, S. R., Kraut, R. E., Lerch, F. J., Scherlis, W. L., McNally, M. M., and Cadiz, J. J. Coordination, overload and team performance: effects of team communication strategies. In *Proc. of the 1998 ACM conference on Computer supported cooperative work* (New York, NY, USA, 1998), ACM, pp. 275–284.

[50] Gansner, E. R., and North, S. C. Improved force-directed layouts. In *Proceedings of the 6th International Symposium on Graph Drawing* (1998), GD '98, pp. 364–373.

[51] Garrison, D. R. Online collaboration principles. *Asynchronous Learning Networks 10* (2006).

[52] Ghoniem, M., Fekete, J.-D., and Castagliola, P. A comparison of the readability of graphs using node-link and matrix-based representations. In *Information Visualization, 2004. INFOVIS 2004. IEEE Symposium on* (2004), pp. 17 –24.

[53] Girvan, M., and Newman, M. Community structure in social and biological network. In *Natl. Acad. Sci.* (2002), vol. 99, pp. 7821–7826.

[54] Görke, R., Gaertler, M., and Wagner, D. Lunarvis–analytic visualizations of large graphs. In *Graph Drawing* (2008), Springer, pp. 352–364.

[55] Gotsman, C., and Koren, Y. Distributed graph layout for sensor networks. In *Graph Drawing* (2005), Springer, pp. 273–284.

[56] Greffard, N., Picarougne, F., and Kuntz, P. Visual community detection: an evaluation of 2d, 3d perspective and 3d stereoscopic displays. In *Graph Drawing* (2012), Springer, pp. 215–225.

[57] Gretarsson, B., Bostandjiev, S., ODonovan, J., and Höllerer, T. Wigis: a framework for scalable web-based interactive graph visualizations. In *Graph Drawing* (2010), Springer, pp. 119–134.

[58] Grimstead, I. J., Walker, D. W., and Avis, N. J. Collaborative visualization: A review and taxonomy. In *Proc. of the 9th IEEE International Symposium on Distributed Simulation and Real-Time Applications* (Washington, DC, USA, 2005), IEEE Computer Society, pp. 61–69.

[59] Gronemann, M., and Jünger, M. Drawing clustered graphs as topographic maps. *Graph Drawing E-pring Archive* (2012).

[60] Groth, D., and Skandier, T. *Network Study Guide: Exam N10-003.* Sybex, May 2005.

[61] Gutwin, C., and Greenberg, S. The mechanics of collaboration: Developing low cost usability evaluation methods for shared workspaces. In *Proc. of the 9th IEEE International Workshops on Enabling Technologies* (2000), pp. 98–103.

[62] Hachul, S., and Jünger, M. Drawing large graphs with a potential-field-based multilevel algorithm. In *12th Int. Symp. on Graph Drawing* (2004), vol. 3383, pp. 285–295.

[63] Han, K., Ju, B.-H., and Park, J. Interviewer: Dynamic visualization of protein-protein interactions. In *Graph Drawing* (2002), Springer, pp. 27–49.

[64] Harel, D., and Koren, Y. Graph drawing by high-dimensional embedding. In *Revised Papers from the 10th International Symposium on Graph Drawing* (2002), GD '02, pp. 207–219.

[65] Heer, J., and Agrawala, M. Design considerations for collaborative visual analystics. *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology* (2007), 171–178.

[66] Heider, F. Attitudes and cognitive organization. *Journal of Psychology 21* (1946), 107–112.

[67] Henry, N., and Fekete, J.-D. Matrixexplorer: A dual-representation system to explore social networks. *IEEE Transactions on Visualization and Computer Graphics 12*, 5 (2006), 677–684.

[68] Holten, D. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics 12* (September 2006), 741–748.

[69] Hong, S.-H., and Murtagh, T. Visualisation of large and complex networks using polyplane. In *Graph Drawing* (2005), pp. 471–481.

[70] Hu, X., Lu, A., and Wu, X. Spectrum-based network visualization for topology analysis. *Computer Graphics and Application* (2012).

[71] Hu, X., Song, H., Harrison, L., Lu, A., Gao, J., and Wang, W. Towards effective collaborative analysis for distributed intrusion detection. In *The 6th IASTED International Conference on Human-Computer Interaction* (2011).

[72] Huffaker, B., Nemeth, E., and Claffy, K. Otter: A general-purpose network visualization tool. In *International Networking Conference (INET)* (1999).

[73] Indiramma, M., and Anandakumar, K. Collaborative decision making framework for multi-agent system. In *Computer and Communication Engineering, 2008. ICCCE 2008. International Conference on* (may 2008), pp. 1140–1146.

[74] Isenberg, P., and Carpendale, S. Interactive tree comparison for co-located collaborative information visualization. *IEEE Transactions on Visualization and Computer Graphics 13*, 6 (2007), 1232–1239.

[75] Isenberg, P., Fisher, D., Paul, S. A., Ringel Morris, M., Inkpen, K., and Czerwinski, M. Co-located collaborative visual analytics around a tabletop display. *IEEE Transactions on Visualization and Computer Graphics 18*, 5 (May 2012), 689–702.

[76] Itoh, T., Takakura, H., Sawada, A., and Koyamada, K. Hierarchical visualization of network intrusion detection data. *Computer Graphics and Applications, IEEE 26*, 2 (2006), 40–47.

[77] Jankun-Kelly, T. J., and Ma, K.-L. Moiregraphs: radial focus+context visualization and interaction for graphs with visual nodes. In *Proceedings of the Ninth annual IEEE conference on Information visualization* (2003), INFOVIS'03, pp. 59–66.

[78] Johnson, B., and Shneiderman, B. Tree-maps: A space-filling approach to the visualization of hierarchical information structures. In *Visualization, 1991. Visualization'91, Proceedings., IEEE Conference on* (1991), IEEE, pp. 284–291.

[79] Johnson, C., and Hansen, C. *Visualization Handbook.* Academic Press, Inc., Orlando, FL, USA, 2004.

[80] Kermarrec, A.-M., and Moin, A. Energy Models for Drawing Signed Graphs. Rapport de recherche, INRIA, 2011.

[81] Kermarrec, A.-M., and Thraves, C. Can everybody sit closer to their friends than their enemies? In *Proceedings of the 36th international conference on Mathematical foundations of computer science* (2011), MFCS'11, pp. 388–399.

[82] Klein, G. Features of team coordination. *New Trends in Cooperative Activities:Understanding System Dynamics in Complex Environments* (2001), 68–95.

[83] Koren, Y., Carmel, L., and Harel, D. Drawing huge graphs by algebraic multigrid optimization. In *Multiscale Modeling and Simulation* (2003), vol. 2697, pp. 645–673.

[84] Koren, Y., and Çivril, A. The binary stress model for graph drawing. In *Graph Drawing* (2009), Springer, pp. 193–205.

[85] Kraut, R. E., and Attewell, P. Media use in a global corporation:electronic mail and organizational knowledge. *In S.Kiesler(Ed.) Research milestones on the information highway.* (1997).

[86] Kunegis, J., Lommatzsch, A., and Bauckhage, C. The slashdot zoo: mining a social network with negative edges. In *Proceedings of the 18th international conference on World wide web* (2009), WWW '09, pp. 741–750.

[87] Kunegis, J., Schmidt, S., Lommatzsch, A., Lerner, J., De, E. W., and Albayrak, L. S. Spectral analysis of signed graphs for clustering, prediction and visualization. In *SIAM Int. Conf. on Data Mining* (2010), pp. 559–570.

[88] Lamping, J., and Rao, R. The hyperbolic browser: A focus+ context technique for visualizing large hierarchies. *Journal of visual languages and computing 7*, 1 (1996), 33–55.

[89] Lazos, L., and Poovendran, R. Serloc: Robust localization for wireless sensor networks. *ACM Trans. Sen. Netw. 1*, 1 (2005), 73–100.

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

[91] Livnat, Y., Agutter, J., Moon, S., Erbacher, R. F., and Foresti, S. A visualization paradigm for network intrusion detection. In *Proceedings of the IEEE Information Asssurance Workshop* (2005), pp. 92–99.

[92] Lu, A., Wang, W., Dnyate, A., and Hu, X. Sybil attack detection through global topology pattern visualization. *Information visualization* (2010).

[93] Ma, K.-L., and Wang, C. Social-aware collaborative visualization for large scientific projects. In *International Symposium on Collaborative Technologies and Systems (CTS)* (2008), pp. 190–195.

[94] MacEachren, A., Dai, X., Hardisty, F., Guo, D., and Lengerich, G. Exploring high-d spaces with multiform matrices and small multiples. In *Processdings of the international symposium on information visualization* (2003), pp. 31–38.

[95] MathWorks. Documentation center for mathworks. http://www.mathworks.com/help/stats/hierarchical-clustering.html.

[96] Muelder, C., and Ma, K.-L. Visualization of sanitized email logs for spam analysis. In *Proceedings of APVIS* (2007).

[97] Muelder, C., and Ma, K.-L. Rapid graph layout using space filling curves. *IEEE Transactions on Visualization and Computer Graphics 14* (2008), 1301–1308.

[98] Neale, D. C., Carroll, J. M., and Rosson, M. B. Evaluating computer-supported cooperative work: Models and frameworks. In *Proc. of the 2004 ACM Conference on Computer Supported Cooperative Work* (2004), pp. 112–121.

[99] Newman, M. E. J. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E 74* (Sep 2006), 036104.

[100] Newsome, J., Shi, E., Song, D., and Perrig, A. The sybil attack in sensor networks: analysis & defenses. In *Proc. of International Symposium on Information processing in sensor networks* (2004), pp. 259–268.

[101] Nguyen, Q., Hong, S.-H., and Eades, P. Tgi-eb: A new framework for edge bundling integrating topology, geometry and importance. In *Graph Drawing* (2012), Springer, pp. 123–135.

[102] Nguyen, Q. V., and Huang, M. L. A space-optimized tree visualization. In *Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)* (2002), INFOVIS '02, pp. 85–92.

[103] Noack, A. An energy model for visual graph clustering. In *Proceedings of the 11th International Symposium on Graph Drawing (GD 2003), LNCS 2912* (2003), Springer-Verlag, pp. 425–436.

[104] Noss, R., Hoyles, C., Gurtner, J.-L., Adamson, R., and Lowe, S. Face-to-face and online collaboration: appreciating rules and adding complexity. *International journal of Continuing Engineering Education and Lifelong Learning 12* (2002), 521–540.

[105] Park, K. S., Kapoor, A., and Leigh., J. Lessons learned from employing multiple perspectives in a collaborative virtual environment for visualizing scientific data. *Proceeding of Collaborative Virtual Environments* (2000), 73–82.

[106] Pupyrev, S., Nachmanson, L., Bereg, S., and Holroyd, A. Edge routing with ordered bundles. In *Graph Drawing* (2012), Springer, pp. 136–147.

[107] Pupyrev, S., Nachmanson, L., and Kaufmann, M. Improving layered graph layouts with edge bundling. In *Graph Drawing* (2011), Springer, pp. 329–340.

[108] Qu, H., Zhou, H., and Wu, Y. Controllable and progressive edge clustering for large networks. In *Graph Drawing* (2007), Springer, pp. 399–404.

[109] Raitner, M. Hgv: A library for hierarchies, graphs, and views. In *Graph Drawing* (2002), Springer, pp. 361–382.

[110] Reingold, E. M., and Tilford, J. S. Tidier drawings of trees. *IEEE Trans. Softw. Eng. 7* (1981), 223–228.

[111] Rekimoto, J., and Green, M. The information cube: Using transparency in 3d information visualization. In *Proceedings of the Third Annual Workshop on Information Technologies & Systems* (1993), pp. 125–132.

[112] Ren, P., Gao, Y., Li, Z., Chen, Y., and Watson, B. Idgraphs: Intrusion detection and analysis using stream compositing. *IEEE Comput. Graph. Appl. 26*, 2 (2006), 28–39.

[113] Robertson, G. G., Mackinlay, J. D., and Card, S. K. Cone trees: animated 3d visualizations of hierarchical information. In *Proceedings of the SIGCHI conference on Human factors in computing systems: Reaching through technology* (1991), pp. 189–194.

[114] Sallaberry, A., Muelder, C., and Ma, K.-L. Clustering, visualizing, and navigating for large dynamic graphs. In *Graph Drawing* (2013), Springer, pp. 487–498.

[115] Seary, A., and Richards, W. Spectral methods for analyzing and visualizing networks: An introduction. In *National Resrach Council, Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers* (2003), pp. 209–228.

[116] Shipman, S., and Wholey, J. Performance measurement and evaluation: Definitions and relationships, April 1998.

[117] Shneiderman, B., and Aris, A. Network visualization by semantic substrates. *Visualization and Computer Graphics, IEEE Transactions on 12*, 5 (2006), 733–740.

[118] Stasko, J., and Zhang, E. Focus+ context display and navigation techniques for enhancing radial, space-filling hierarchy visualizations. In *Information Visualization, 2000. InfoVis 2000. IEEE Symposium on* (2000), pp. 57–65.

[119] Suh, B., Chi, E. H., Pendleton, B. A., and Kittur, A. Us vs. them: Understanding social dynamics in wikipedia with revert graph visualizations. In *Proceedings of the 2007 IEEE Symposium on Visual Analytics Science and Technology* (2007), VAST '07, pp. 163–170.

[120] Suntinger, M., Obweger, H., Schiefer, J., and Groller, M. E. Event tunnel: Exploring event-driven business processes. *Computer Graphics and Applications, IEEE 28*, 5 (2008), 46–55.

[121] Sybex. *Networking Complete 3RD Edition*. Sybex, October 2002.

[122] Szell, M., Lambiotte, R., and Thurner, S. Multirelational organization of large-scale social networks in an online world. *Proceedings of the National Academy of Sciences (PNAS)* (2010).

[123] Tanaka, Y., Okada, Y., and Niijima, K. Treecube: Visualization tool for browsing 3d multimedia data. In *Proceedings. Seventh International Conference on Information Visualization, 2003. IV 2003* (2003), pp. 427–432.

[124] Tee Teoh, S., and Kwan-Liu, M. Rings: A technique for visualizing large hierarchies. In *Graph Drawing* (2002), Springer, pp. 51–73.

[125] Torgeson, W. Multidimensional scaling of similarity. *Psychometrika 30* (1965), 379–393.

[126] Trefethen, L. N., and Bau, D. *Numerical Linear Algebra*. SIAM: Society for Industrial and Applied Mathematics, 1997.

[127] van Ham, F., Schulz, H.-J., and Dimicco, J. Honeycomb: Visual analysis of large scale social networks. *Human-Computer Interaction–INTERACT 2009* (2009), 429–442.

[128] Waldner, M., Lex, A., Streit, M., and Schmalstieg., D. Design considerations for collaborative information workspaces in multi-display environments. *Proc. of Workshop on Collaborative Visualization on Interactive Surfaces* (2009).

[129] Wang, W., and Lu, A. Visualization assisted detection of sybil attacks in wireless networks. In *Proceedings of ACM Workshop on Visualization for Computer Security (VizSEC)* (2006), pp. 51–60.

[130] Wang, Y., Chakrabarti, D., Wang, C., and Faloutsos, C. Epidemic spreading in real networks: An eigenvalue viewpoint. *Reliable Distributed Systems, IEEE Symposium on 0* (2003), 25.

[131] Wu, L. Spectral analysis of rich network topology in social networks". *Ph.D. dissertation, UNC Charlotte* (2013).

[132] Wu, L., Ying, X., Wu, X., Lu, A., and Zhou, Z.-H. Spectral analysis of k-balanced signed graphs. In *Proceedings of the 15th Pacific-Asia conference on Advances in knowledge discovery and data mining - Volume Part II* (2011), PAKDD'11, pp. 1–12.

[133] Wu, L., Ying, X., Wu, X., Lu, A., and Zhou, Z.-H. Examining spectral space of complex networks with positive and negative links. *IJSNM 1*, 1 (2012), 91–111.

[134] Wu, L., Ying, X., Wu, X., and Zhou, Z.-H. Line orthogonality in adjacency eigenspace with application to community partition. In *IJCAI* (2011), T. Walsh, Ed., pp. 2349–2354.

[135] Yang, B., Cheung, W., and Liu, J. Community mining from signed social networks. *Knowledge and Data Engineering, IEEE Transactions on 19*, 10 (2007), 1333 –1348.

[136] Yang, M. C., and Jin, Y. An examniation of team effectiveness in distributed and co-located engineering teams. *International Journal of Engineering Education 24*, 2 (2008), 400–408.

[137] Ying, X., Wu, L., and Wu, X. A spectrum-based framework for quantifying randomness of social networks. *Knowledge and Data Engineering, IEEE Transactions on 23*, 12 (dec. 2011), 1842 –1856.

[138] Ying, X., and Wu, X. Randomizing social networks: a spectrum preserving approach. In *In the Proceedings of the 8th SIAM Conference on Data Mining* (2008), pp. 739–750.

[139] Ying, X., and Wu, X. On randomness measures for social networks. In *Proceedings of the 9th SIAM Conference on Data Mining* (2009), pp. 709–720.

[140] Yurcik, W. Visflowconnect-ip: A link-based visualization of netflows for security monitoring. In *18th Annual FIRST Conference on Computer Security Incident Handling* (2006).