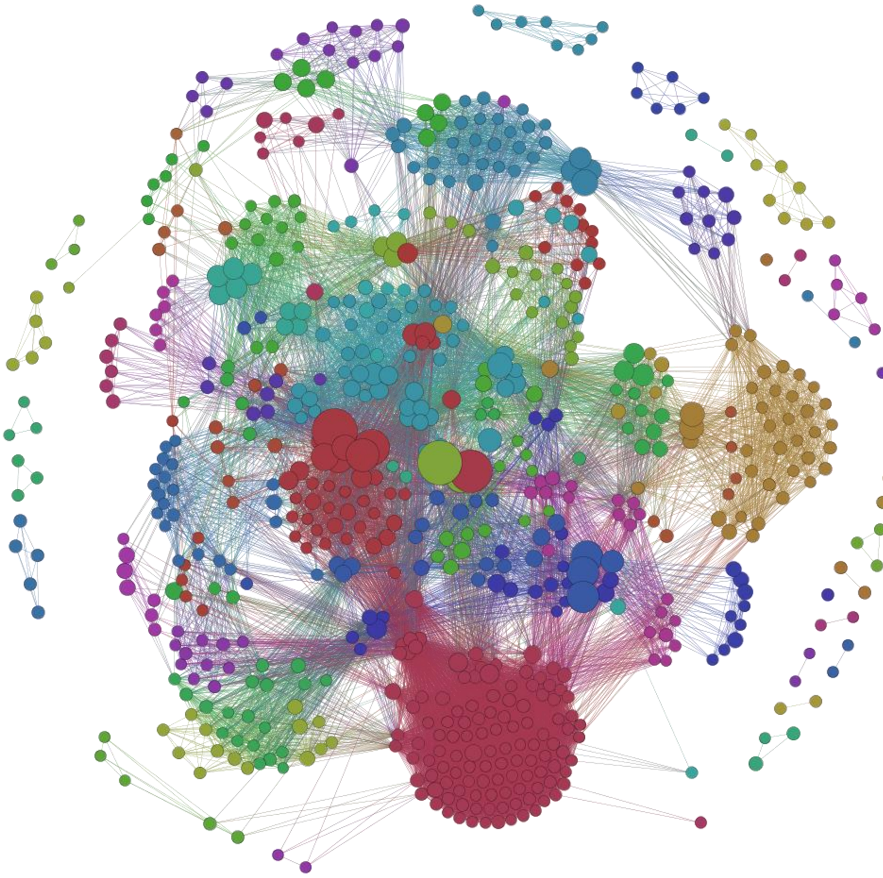


On the Characterization and Comparison of Complex Networks

Sarvenaz Choobdar

Programa Doutoral em Informática das Universidades do Minho, Aveiro e Porto

Departamento de Ciência de Computadores

2015

Orientador

Fernando Manuel Augusto Silva, Professor Catedrático, Faculdade de Ciências da Universidade do Porto

Coorientador

Pedro Manuel Pinto Ribeiro, Professor Auxiliar Convidado, Faculdade de Ciências da Universidade do Porto

University of Porto
Faculty of Science
Department of Computer Science

On the Characterization and Comparison of Complex Networks

Thesis submited:
to doctoral Program in Computer Science of the Universities of Minho, Aveiro and Porto (MAP-i) in partial fulfillment of the requirements for the Degree of Doctor of Philosophy.

Sarvenaz Choobdar

Porto, Jan. 2015

This thesis is realized under Supervision of

Fernando Manuel Augusto Silva

Professor of the
Faculty of Science, University of Porto

and co-supervision of

Pedro Manuel Pinto Ribeiro

Invited Assistant Professor of the
Faculty of Science, University of Porto

This thesis is dedicated to
my mother, who taught me to pursue my dreams
and to
my dear Ali, who taught me how to achieve my dreams.

Acknowledgements

The completion of this thesis is due to many inspiring individuals that I would like to thank them here.

I am extremely fortunate to have my PhD advisor professor Fernando Silva, who gave me this opportunity to start my PhD. I wholeheartedly thank him for inspiring me towards studying networks, for the fruitful research discussions, for all the time he was able to fit into his schedule when I needed, and for many other inspiring things I cannot find enough space to list here. I am also very thankful to my co-adviser Dr. Pedro Ribeiro whose collaboration was essential and fruitful for initiation and completion of my thesis.

I would like to thank professor Maria Eduarda Silva for insightful comments in my work in this thesis and for many motivating discussions. I am also grateful to professor Srinivasan Parthasarthy for receiving me at his research lab in the Ohio State University. Working closely with him and his group opened new ways to my PhD research and led to enrichment of my work.

I would like to thank CRACS/INESC-Porto L.A. and FCT for financially supporting me on my PhD.

I thank my office-mate and friends at CRACS for their generous supports and friendships, we have a great time. A special thanks goes to Sylwia Bulga, who was not only a good colleague but was a true friend and made doing the PhD more joyful for me.

Most of all, I am grateful to my family. My mother ShirinJan Safaie who has been so selfless in supporting me at all stages of my life and career and my siblings who have been encouraging me in the direction of my career. At last but definitely not least, I want to thank my love of life, my husband Ali Marjovi who has inspired me with his hardworking, courageous, enthusiastic, and always cheerful character in life. His support is much appreciated and has led to many interesting and insightful discussions relating to this research.

Abstract

A wide variety of real life structures can be intuitively represented by complex networks. Mining interesting features from these networks is a very important task with an inherent multidisciplinary impact. Past studies have been essentially focusing on single static individual networks. Recently, however, much research effort is geared towards a more dynamic setting, where networks evolve and change over time. By analyzing several instances of a network, we have a more comprehensive set of features that can be used to completely characterize a network. This field is still in its early development stages and making these comparisons is not a trivial task.

Our main goal in this thesis is to provide graph mining techniques geared towards a dynamic setting and capable of discovering core similarities and differences between multiple networks. In order to achieve our research goals, we developed a series of methodologies for *characterization* and *comparison* of networks at two different granularity levels: nodes and subgraphs.

In the first part of this dissertation, we study the network characteristics at node level by casting the nodes into a set of *structural roles*. The structural patterns in the neighborhood of nodes assign unique *roles* to the nodes. Mining the set of existing roles in a network provides a descriptive profile of the network and draws its general picture. The structural role of nodes in a network represent their structural positions and can be associated to functional or organizational roles they may play. We propose methods to: 1) find structural roles and examine how they evolve over time; 2) extract evolutionary roles to represent temporal behavior of nodes; 3) infer pairwise relations for structural roles. We demonstrate the applicability and use of role mining methods in the context of information cascades, and we show how structural roles of users in an information process affect their actions.

In the second part, we develop new methods to characterize and to compare networks at subgraph level by extracting their building blocks (motifs). In this part our focus is on characterizing weighted networks where relations between entities have a certain strength.

We define motifs in weighted graphs as subgraphs that contain unexpected information, and we define new significance measurements to assess their exceptionality. We show how our weighted motif mining approach is useful for comparing and characterizing networks.

We evaluated our methods on a broad range of real data sets, including social and biological networks. In the first part of the thesis, our focus is on social networks and information cascades. We show that topological metrics indeed possess discriminatory power and that different structural patterns correspond to different roles in the process. The extensive experiments demonstrate the efficacy of our structural role mining methods in categorization of users in social activities. In the motif mining part, our focus is more on biological networks, namely gene co-expression networks, for which our proposed definitions of weighted motifs are well suited. The experimental results show that we are able to distinguish between healthy and cancer related tissues, by using exceptionally weighted substructures.

Resumo

Uma grande variedade de estruturas da vida real pode ser representada por redes complexas. Descobrir características interessantes destas redes é por isso mesmo uma tarefa muito importante com um impacto multidisciplinar. No passado, o trabalho de investigação existente focou-se essencialmente em redes individuais estáticas. Recentemente, contudo, tem havido um aumento da investigação direcionada para um ambiente mais dinâmico, onde as redes evoluem e mudam ao longo do tempo. Através da análise de várias instâncias de uma rede, conseguimos ter acesso a um conjunto muito mais compreensivo de dados capazes de caracterizar uma rede. Esta área de investigação está ainda numa fase inicial do seu desenvolvimento e fazer estas comparações não é uma tarefa trivial.

O nosso principal objectivo nesta tese é providenciar um conjunto de técnicas de extração de dados de grafos direcionadas para um ambiente dinâmico e sendo capazes de descobrir as semelhanças e diferenças fundamentais entre múltiplas redes. De modo a atingir os nossos objectivos, desenvolvemos uma série de metodologias para a *caracterização* e *comparação* de redes em dois níveis diferentes de granularidade: nós e subgrafos.

Na primeira parte da dissertação, estudamos as características de uma rede ao nível dos nós, atribuindo-lhes uma função dentro de um possível conjunto de *papeis estruturais*. Os padrões estruturais na vizinhança dos nós atribuem *papeis* únicos a cada um dos nós. Ao descobrir o conjunto de papeis existentes numa rede, obtemos um perfil descritivo que nos permite ter uma ideia geral da rede. O papel estrutural de um nó representa a sua posição estrutural e pode ser associado um papel funcional ou organizacional. Propomos métodos para: 1) descobrir papeis estruturais e examinar como eles evoluem ao longo do tempo; 2) extrair papeis evolucionários que representam o comportamento temporal dos nós; 3) inferir as relações de homofilia para papeis estruturais. É também demonstrada a aplicabilidade da nossa metodologia no contexto da difusão de informação, e mostramos como os papeis estruturais de utilizadores afetam as suas ações durante o fluxo de informação.

Na segunda parte, desenvolvemos métodos para caracterizar e comparar redes ao nível dos subgrafos, extraindo os seus blocos de construção básicos (padrões conhecidos como "motifs"). Nesta parte o nosso foco é na caracterização de redes pesadas onde as relações entre diferentes entidades têm um peso associado. Definimos "motifs" em grafos pesados como sendo subgrafos que contêm informação inesperada, e definimos também novas métricas de significância para aferir esta excecionalidade. Mostramos depois como as nossas ideias podem ser úteis para comparar e caracterizar redes.

Avaliamos todos os nossos métodos num vasto leque de dados reais, incluindo redes sociais e biológicas. Na primeira parte da tese, o foco está nas redes sociais e nos processos de difusão de informação. Mostramos como as métricas topológicas possuem de facto poder discriminatório e que diferentes padrões estruturais correspondem a diferentes partes do processo. A extensa experimentação demonstra a eficácia dos nossos métodos de extração de papeis estruturais na categorização de utilizadores em atividades sociais. Na parte ligada à descoberta de "motifs, o nosso foco está mais nas redes biológicas, em particular nas redes de co-expressão de genes, nas quais se encaixam muito bem as definições de padrões pesados propostas. Os resultados experimentais mostram que somos capazes de distinguir entre tecidos saudáveis e cancerígenos, usando a excecionalidade de subestruturas pesadas.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Graphs are one of the most ubiquitous data structures to model real world network data in different disciplines such as biology or sociology. As the amount of network data is massively increasing, new methods are also required to capture and to provide a better perception of networks. Complex systems can be modeled using graphs in which nodes represent the entities of the system, and edges represent the inter-relationships among the entities. Each edge may be associated with a weight, a positive real number indicating the strength of the relationship being modeled. The edges may also be directed, indicating that the relationship is asymmetrical. Some examples of domains or systems that are amenable to representation as graphs include:

- Networks of web pages, where pages are nodes, and hyperlinks form edges.
- Organizational network of researchers, where nodes represent researchers and two nodes are connected if they are affiliated with the same research unit as depicted in Figure 1.1
- Social networks, where users are nodes, and different relationships, such as friendship, biological relatedness, workplace collaboration or scientific co-authorship are modeled as connections.
- Biological networks, such as protein-protein interaction (PPI) networks or regulatory networks. In PPI networks, nodes are proteins and the interactions between these form the edges, indicating that the corresponding proteins interacted as part of executing a biological process or function. In regulatory networks, the nodes are genes and edges indicate that one gene is regulated by another.

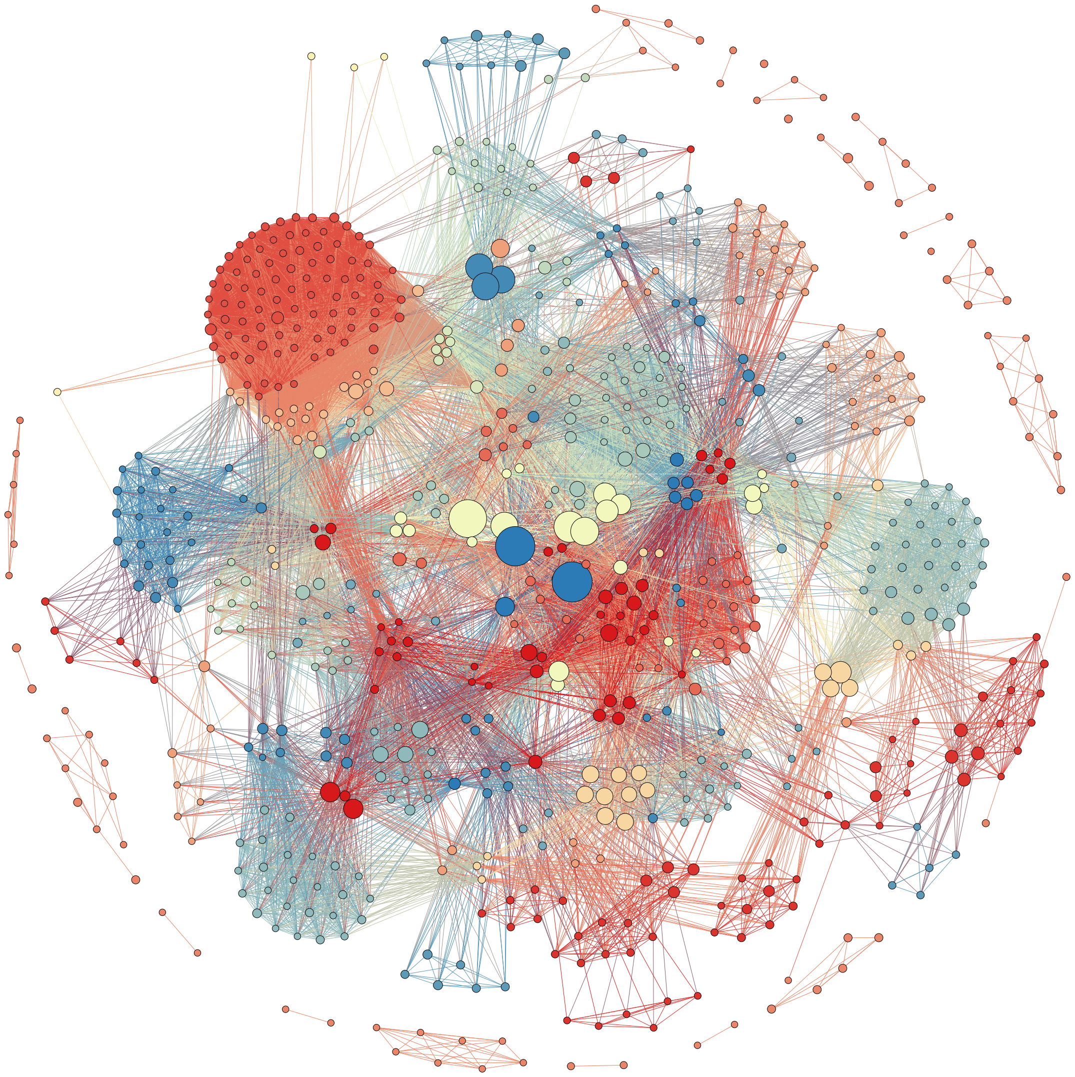


Figure 1.1: Organizational network of Portuguese researchers active in the field of breast cancer. Nodes represent researchers and two nodes are connected if they are affiliated with the same research unit. Nodes are color coded by their structural roles.

Research on complex network data analysis has been very prolific and a large variety of characterization methodologies emerged, such as graph clustering [GN02, Sch07], node classification [MP07], network motifs discovery [MSOI$^+$02] or frequent subgraph mining [YH02a, HWP03a]. All these methods share one goal: discovery of regularities in data in the form of connectivity patterns that characterize the underlying graphs. Most real world networks are complex, in the sense that they present non trivial topological features. Although the vast majority of complex networks have some common regularities such as presenting a small diameter [Bar14, AJB99] or a scale free structure [New05, FFF99], every individual network has some intrinsic unique distinguishable characteristics.

The study of networks is becoming more and more application dependent, given that truly general patterns and regularities are not very informative for specific applications. For some domains such as social networks, the study of individuals and their properties can be more revealing about the whole system. However, in other applications such as biological networks, the study of groups of nodes can better characterize the patterns in the network.

In this thesis we are interested in methods to enable network characterization. To pursue this goal we study networks at two different granularity levels:

1. node level: we study nodes properties in a *role mining* framework.
2. subgraph level: we study connectivity patterns of groups of nodes in a *motif mining* framework.

The structure of a network is determined by its links or connections. By observing the links, one can derive a set of features that characterize the structural position of each node, which can in turn be used to help identify its role. This type of structural role mining has applications in many domains. For example, in the case of online social networks, it is important to know the users' position in the network in order to create personalized marketing campaigns. Another example is viral marketing, where the structural role of users is essential in targeting the appropriate users in order to achieve maximum coverage of the network, to spread ideas such as ads or news. In fact, *structural roles* are gaining increasing attention in recent years, and they are now used as a tool for tasks such as node classification [ZWY$^+$13], identity resolution [HGER$^+$12, GERD13], exploratory network analysis [HGER$^+$12] and anomaly detection [RNGH13]. Role mining derives a profile of a network at microscopic level that can be used in many network applications, namely to study the network structure and for node classification. Since the set of roles in a network represents general existing behaviors, it can also be used to detect outliers, that is, to detect nodes that deviate significantly from existing roles.

In many complex networks such as biological networks, the study of groups of nodes in the form of substructures (subgraphs) is more informative and fact-revealing than the study of individual nodes in the networks. For example in microarray data analysis, finding individual differently expressed genes cannot reveal key biological functions or processes associated with each disease. Instead, gene modules are studied in order to characterize different gene co-expression networks across different types of tumor biopsy samples. A pattern in a network is normally defined as a subgraph which is very frequent or infrequent (in case of anomalies). A specific form of patterns are called motifs, which can be thought of as small subgraphs that appear in a network at significantly higher frequencies than what would be expected in similar randomized networks [MSOI$^+$02]. This type of patterns can help in characterizing the networks since they are not frequent only by chance and, therefore, significantly highlight the specific structural properties of the networks. This is why motifs are also known as the building blocks of networks. It has been demonstrated that they can have functional significance in transcriptional regulatory networks [SOMMA02] or protein-protein interaction networks [AA04]. Figure 1.2 demonstrates application of motif mining in characterizing gene co-expression networks.

To better motivate our work and also show the importance and helpfulness of these methods for network characterization and comparison, we start by describing their potential use in some applications for role and motif mining in networks. Then, we introduce the main goals of this thesis as well as the research questions tackled to achieve these goals. This is followed by a summary of the thesis's contributions, which are, in turn, our proposed solutions for the raised research questions. Finally, we describe the remainder of the thesis in terms of content organization and corresponding publications produced during this research.

1.1 Motivation

Advances in technology made complex networks ubiquitous. They are present everywhere, namely in social, biological or technological networks. Since networks represent data entities and its relationships, there is an ever increasing need for new methods to explore and understand these new sources of useful information. In particular, we list a number of applications where discovering connectivity patterns in the form of structural roles or motif profiles can help in better characterizing the networks.

Networks Dynamics: An important aspect of complex networks is the temporal dimension, which has been studied from different angles such as community evolution [LCZ$^+$08, APU07], graph growth models [LKF05b] and link prediction [LNK07]. Role based analysis

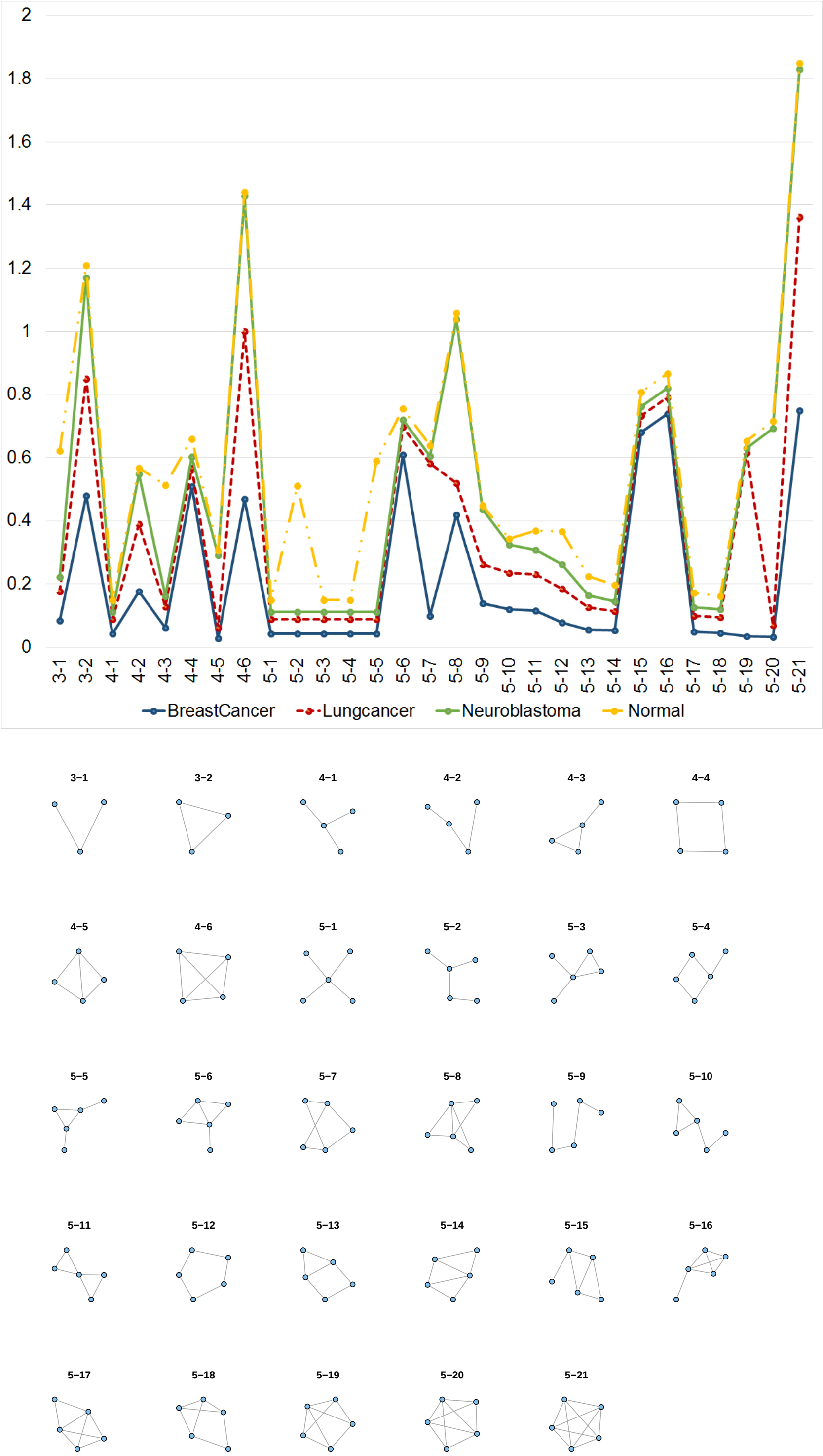


Figure 1.2: Motif profiles of gene co-expression networks of different types (left plot). All subgraphs from size 3 to 5, normally used for motif discovery (right plot).

of networks is another aspect of network dynamic study that depict networks evolution from a microscopic point of view. In a large dynamic network, the temporal structural behaviors of individual nodes can be learned by structural role mining which identifies unusual activities or patterns. For instance, in an IP-to-IP network, we may want to learn the “behavioral roles” of individual hosts and monitor their changes over time. This would allow us to characterize the dynamic behaviors of individual hosts and also detect when a machine or host becomes compromised, or begins having unusual behaviors with respect to the global network dynamics. Rossi and Gallagher defined temporal structural roles as a combination of similar structural features that were learned from the initial network. Since similar structural properties are combined into a single role, then each role represents a different structural pattern (or connectivity pattern) [RGNH12]. In this thesis, we follow the same definition of dynamic roles, however we propose dynamic role mining methods based of clustering algorithm instead of block models.

Node Classification: In some complex networks, a subset of the nodes have labels such as demographic values, interests, beliefs or other characteristics of the nodes (users). Node classification involves determining the label of a node in a network that is partially labeled. Normally, it is assumed that some of the nodes have a predefined label and the labels for the rest of the nodes are predicted using relational classifiers [TAK02, BCM11]. Commonly, labels of nodes may fulfill specific roles. For example, in a Twitter network, users can be identified as an advertiser, a content contributor, or an information receiver. In LinkedIn, users can be associated with different professional roles such as engineer, salesperson, or a recruiter. Previous research work mainly focuses on using categorical and textual information to predict the attributes of users. However, it cannot be applied to a large number of users in real social networks, since much of such information is missing, possibly outdated and non-standard. The structural position of people in online social networks is quantitatively correlated to their actions [RTU13]. The network characteristics reflect the social situations of users in an online society and can be used as predictors for node classification. In a supervised setting, Zhao et al. [ZWY$^+$13] used structural properties in combination with demographic features to predict social statuses of users in a network. However, in this thesis we merely rely on structural properties to classify users.

Networks Comparison: Comparing a diseased cellular network to a healthy one might bring new insight to determine a cure for a disease [MNHP10, BL06]. Similarly to sequence comparison, biological networks across species can be compared against each other to determine common substructures which may be the reason of their equivalent functionality; but how can we efficiently provide a meaningful measure of structural similarity (or distance)? Such measures are extremely useful for numerous graph-mining tasks. One such task is clustering: given a set of graphs, find groups of similar ones; conversely, find anomalies or discontinuities, i.e., graphs that stand out from the rest [BKERF12]. A motif profile of a network acts as the "signature vector" that can be used to discover similarities between networks. In addition, structural role configuration in networks can be another reference point for comparison.

1.2 Goals and research questions

In this thesis we study connectivity patterns in complex networks at two different levels: nodes and subgraphs.

1. Role mining (node level): we design a series of methods for role mining in social networks to model roles of nodes in a network regarding different parameters, including network dynamic and pairwise dependence.
2. Motif mining (subgraph level): we design new methods for motif mining in weighted graphs. An important dimension of complex networks is embedded in the weights of its edges. Incorporating this source of information on the analysis of a network can greatly enhance our understanding of it. We study how motif profiles can classify biological networks across different types of tumor biopsy samples.

Each level of network characterization pursues different research questions as described in the following subsections.

1.2.1 Role mining

We are interested in exploring which structural properties of nodes better distinguish nodes within the network and can form a good feature vector for classifying nodes into *structural roles*. We examine how temporal behavior of nodes impacts their roles. We also study if structural roles of nodes depend on the neighbors' roles. The first part of this thesis is dedicated precisely to these topics, and we look at the general problem of role mining, provide a framework for its study. We address the following questions:

- *How are structural roles formed and changed over time? When do significant changes occur in the role configuration?* Although some recent work has focused on the analysis of dynamic networks [BHKL06, CMG09a, CKT06, LKF05b, PS11], there has been less research on developing models of temporal behavior in large scale network datasets. Yang and Leskovec [YL11] used the temporal link and attribute patterns to improve predictive models. In addition, there are some works on identifying clusters in dynamic data [SFPY07] but these methods focus on discovering underlying communities of nodes that are densely connected together over time. In contrast, we are interested in uncovering the behavioral patterns of nodes in the graph and modeling how those patterns change over time.

- *Does the role of a node dependent on its neighbors'? Do users at a similar structural position tend to connect to each other?* Pairwise dependencies (homophiliy), i.e the tendency of users to connect with users of similar interest and social demography [MSLC01], is one of the sources of information for user behavior modeling [SR08, LAH07] and user classification [ZWY$^+$13]. However this is yet an open question for structural roles.

- *How is the temporal behavior of nodes reflected in their structural roles? How can we detect dynamic roles of nodes?* We formulate and study the problem of evolutionary role extraction where a sequence of graph snapshots are given and the goal is to find the roles of active nodes at the current time. These roles must reflect the structure of the network at the current time and must be consistent with existing roles in the network, extracted at previous times. The evolutionary role extraction must fulfill the following two tasks: 1) the role of nodes at current time should be close to previous time, if the connectivity of nodes does not deviate from previous time points; 2) the set of roles must be modified to reflect the new structure, if the structure of the network changes significantly.

- *What is the role mining application in social networks? Do structural roles of users reflect their social roles in a social network?* We study information propagation of *stories* in social networks, and we concentrate on the effects of structural patterns on two different properties: level of *influence* and *blockage rate*. We categorize users into different roles in a social activity from these two points of view. User influence is related to the cascade size a user can cause, that is, the amount of other users that receive stories propagated by such cascade. Blockage rate amounts to the number of stories a user does not repost, normalized to the total number of received stories. We use network characteristics of users to classify them into social groups and try to find a correspondence between topological positions and social roles.

1.2.2 Motif mining

Our focus in this part is to extract motifs as building blocks of weighted networks. We are interested in understanding how edge weights can be used in the process of motif mining. We examine which significance score can detect outstanding patterns in the networks by incorporating weights. In particular, we exploit information theory to assess exceptionality of subgraphs. The main questions of interest in this part of the thesis are:

- *Given a large graph with weights over the edges, how can we find outstanding substructures? What function can measure significance of a pattern regarding the weights other than the frequency?* Unexpectedly frequent subgraphs, known as motifs, can help in characterizing the structure of complex networks. Most of the existing methods for finding motifs are designed for unweighted networks, where only the existence of a connection between nodes is considered, and not its strength or capacity. However, in many real world networks, edges contain more information than just simple node connectivity.

- *How can motif mining help network comparison? Do the motif profiles of networks distinguish them? Particularly in biological networks? How different is a healthy network from disease associated one?* Incorporating weight information on the analysis of a network can greatly enhance our understanding of it. This is the case for gene co-expression networks (GCNs), which encapsulate information about the strength of correlation between gene expression profiles. Classical unweighted GCNs use thresholding for defining connectivity, losing some of the information contained in the different connection strengths. One important goal of studying GCNs is to predict gene functions and disease biomarkers such as the discovery of cancer related genes [PHS$^+$07, ZHXJ09]. Here we are interested in studying the structure of networks across healthy tissues and cancer related ones. We want to understand how a healthy network looks like and what makes it different from an unhealthy one, and what distinguishes GCNs across different diseases, such as distinct cancer types. Are there subnetworks (groups of densely connected nodes in the network) in cancer sample networks that does not appear in healthy networks?

1.3 Contributions and thesis organization

The contributions of this thesis are twofold, and are based on the aforementioned goals: 1) role mining 2) motif mining. Accordingly, we organize the thesis into two parts:

1.3.1 Role mining

We develop a collection of novel methods to discover structural roles in a graph for various settings: when the graph structure is evolving and the evolution of roles from one time to another is of interest; when the graph is changing over time and roles reflect the dynamics of nodes as well as structural positions; and when pairwise dependence between structural roles is considered as well. The collection of these methods provides a package for role mining under different circumstances. Our contribution to this part can be summarized as follows.

- **Structural role mining and tracking**: We developed a methodology for extracting roles in the networks and monitor the evolution of roles over time. We describe this evolution by defining a set of events and extracting the transition patterns. We define a set of events to explain the evolution of roles in the networks. (Chapter 3)
- **Pairwise structural role mining in social networks**: We study the patterns of pairwise dependency for structural roles, showing that pairwise dependencies can improve discovery of some roles, while for others can be misleading. We show that to accurately infer a role, we cannot propagate role labels through all connections. We develop a new probabilistic relational framework called *SR-diffusion* to jointly model pairwise dependence and structural positions of users. We design an algorithm to learn the SR-diffusion model in social networks, where the hidden variable role is inferred regarding the observed variables of ego properties of users and connections. We define a cost function to model the pairwise dependencies and structural similarities. This algorithm, iteratively infers the social roles of users based on structural similarities in the network and by propagating roles through connections. (Chapter 4)
- **Evolutionary structural role mining in complex networks**: We formulate a framework for evolutionary role mining in complex networks. We design and evaluate a new weighted clustering ensemble to dynamically learn tempo-structural roles of nodes in a dynamic network. Through experiments on real word datasets, we show that our method is better capable to separate nodes into their structural roles and the

discovered roles set is more coherent to historical behavior of user, comparing to baseline methods and counterpart methods in the literature. (Chapter 5)

- **Social roles inference in information cascades**: An in-depth analysis of how pure topological features are related to the roles of users in information cascades, namely their influence and blockage rate. We show how information cascade modeling can benefit from role mining by predicting influential users in information cascade from the role membership of users. (Chapter 6)

1.3.2 Motif mining

We are among the first to propose a method for motif mining in graphs so that weight information on edges is captured as well. (Chapter 7)

- **Motif mining in weighted networks**: We developed state of the art methods for motif discovery in graphs with weights. We define a subgraph as a motif if the weights of edges inside the subgraph hold a significantly different distribution than what would be found in a random distribution. We use an information theoretic measure to calculate the significance score of the subgraph, avoiding the time consuming generation of random networks to determine statistic significance. (Chapter 8)
- **Discovering biomarkers in gene co-expression networks**: We compare gene co-expression networks of normal tissues and cancer associated ones by their motif profiles. We show that our weighted motif definition is capable of distinguishing networks by their types. Using gene ontology terms enrichment analysis, we demonstrate predictability of weighted motifs in classifying functionality of the disease-associated genes. (Chapter 8)

1.4 Bibliographic note

Parts of this thesis are published in the following papers:

- Sarvenaz Choobdar, Pedro Ribeiro, Srinivasan Parthasarathy and Fernando Silva. Dynamic inference of social roles in information cascades. Journal of Data Mining and Knowledge Discovery, Springer, 2015

- Sarvenaz Choobdar, Pedro Ribeiro and Fernando Silva. Discovering Weighted Motifs in Gene co-expression Networks. Proceedings of the 30th ACM/SIGAPP Symposium On Applied Computing, 2015

- Sarvenaz Choobdar, Pedro Ribeiro and Fernando Silva. Querying volatile and dynamic networks. Encyclopedia of Social Network Analysis and Mining, New-York: Springer Science + Business Media, 2014.

- Sarvenaz Choobdar, Pedro Ribeiro and Fernando Silva. Motif Mining in Weighted Networks. Proceedings of the IEEE ICDM Workshop on Data Mining in Networks (DaMNet), 2012.

- Sarvenaz Choobdar, Pedro Ribeiro and Fernando Silva. Event Detection in Evolving Networks. Proceedings of the IEEE International Conference on Computational Aspects of Social Networks (CASoN), 2012.

- Sarvenaz Choobdar, Pedro Ribeiro and Fernando Silva. Comparison of co-authorship networks across scientific fields using motifs. Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), 2012.

- Sarvenaz Choobdar, Fernando Silva and Pedro Ribeiro. Network node label acquisition and tracking. Proceedings of the Portuguese Conference on Artificial Intelligence (EPIA), 2011.

Part I

Structural role mining

Chapter 2

Background

What are structural roles? How structural can roles can reveal complex network characteristics and dynamics?

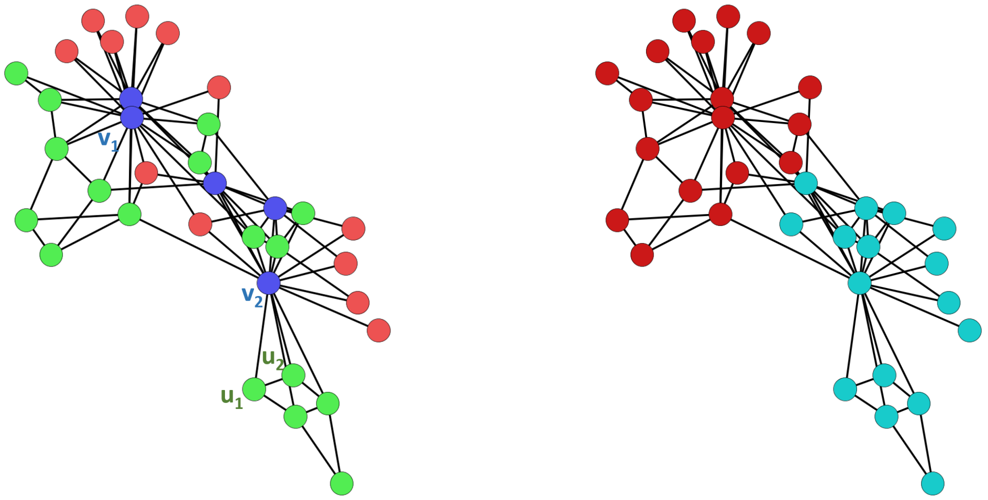
Structural role mining is essentially an exploratory task where no a-priori class for nodes is available and our goal is to assign a role to each of the nodes. A role can be seen as a "class" of nodes that share similar structural properties such as their degree, clustering coefficient or betweeness centrality. Moreover, roles can often be associated with various functional or organizational roles such as being members of a clique, being the center or endpoints of a star, or acting as articulation points that serve as bridges between different parts of a network. Given how ubiquitous networks are, role mining finds applications in many domains. An example is the case of online social network, where a user's position in the network is important for a personalized marketing campaign. More specifically, in viral marketing the structural role is essential for targeting appropriate users in order to attain maximum coverage of the network when spreading information, such as news or advertising campaigns [ALTY08, WLJH10]. Structural roles are therefore gaining an increased importance and are becoming an important tool for tasks such as node classification [ZWY$^+$13], identity resolution [HGER$^+$12, GERD13], exploratory network analysis [HGER$^+$12] and or outlier detection [RNGH13]. The first part of this thesis is precisely dedicated to studying the *role mining* problem. We begin our journey in this chapter, where we introduce the general framework of role mining, reviewing the state of the art and showing some example applications.

2.1 Structural role definition

Role is a relatively vague concept used for describing the function or effect of an object in a system. For example, organizational roles such as managers, or CEOs explain particular responsibilities that a person can hold in a business system [ZWY$^+$13]. Similarly, participation roles such as broadcasters or discussion promoters explain the responsibilities that a user may have in a social media system [BBHM13]. Although role theory is more advanced in sociology, it has been shown that nodes in networks of other domains, such as biology, also present different functional roles.

Given this multidisciplinarity, defining and interpreting roles is really a domain-dependent task and depends on the goal of study. For example, in biological networks the genes responsible for cancer-related functions [GA05, WLY14] are those raising more interest while in social networks users with high influence over their neighborhood are more important for broadcasting information [RTU13, ALTY08]. Regardless of domain specific interpretations, for almost any possible scenario or system we can define a set of roles for the respective entities, representing their impact and involvement in their circle of influence. If we look at the complex networks modeling these systems, we can see that the structural position of nodes is correlated to their role and we can find equivalent classes of nodes as *structural roles* in the networks regarding a set of features. These are called structural roles as since in their discovery only pure topological properties are used, describing the nodes and how edges are connecting them, as opposed to using non-graph features.

Going further into a more formal description, we can think of the role mining task in networks as a process that partitions the nodes into classes of equivalent nodes. This is different from community discovery in networks where the goal is to find highly connected groups of nodes, that is, with few connections to the rest of the network. This difference between role mining and community discovery is highlighted in Figure 2.1. Figure 2.1a shows the structural roles in the network of Karate club [Zac77], and Figure 2.1b demonstrates the existing communities in the network. In the role mining task, all nodes holding the same role should be equivalent under the predefined node equivalence relation. Given a graph $G = (V, E)$, where V is the set of nodes, and E the set of edges, Rossi and Ahmed [RA14] expressed this equivalence relation as $\forall u, v \in V; r(u) = r(v) \Leftrightarrow u \equiv v$, where $r(u)$ and $r(v)$ are the role classes of nodes u and v respectively. This node equivalence relation is very flexible and can be defined using different approaches. For example, two nodes can be considered equivalent if they have connections to exactly the same neighbors. Figure 2.1 exemplifies this concept. Nodes u_1 and u_2 are structurally equivalent as they are only



(a) Colors correspond to different structural roles, blue nodes are hubs (ex. v_1, v_2), red nodes are prepherial and and green nodes are member-of-cliques (ex. u_1, u_2).

(b) Nodes are partitioned into two community, nodes in blue color are more connected together than to the nodes in red color.

Figure 2.1: Structural roles vs community structure on Karate Club network [Zac77].

connected to the same set of nodes, that is, they are both leafs of a star shaped structure. However, this definition is not complete and does not hold for many other nodes, such as v_1 and v_2 in the figure. These nodes are structurally equivalent, with both acting as intermediate nodes between different highly connected regions, even though they do not have the same set of neighbors. This shows us that it is not only the connections themselves that define the structural roles. Instead, we need a more complete set of topological properties to represent the structural position of a node in the network.

Following Rossi and Ahmed [RA14], the strict equivalence of two nodes u and v, based on a set of features $x_1, x_2, ..., x_m$, is defined as:

$$(\forall i \in [1:m] : |x_i(u) - x_i(v)| < \varepsilon) \Rightarrow u \equiv v \tag{2.1}$$

This definition implies that two nodes u and v share the same role, if and only if they have similar feature-values. This definition raises two very important questions regarding role mining. Which set of features can really characterize a node and thus better explain structural roles? And, how can we assign nodes to structural roles? Answering each of these questions depends on a number of parameters and conditions that very much depend on the goal of role mining study and the context. However, the task of role mining constitutes a general framework that we can follow, as we will explain in more detail in the following sections.

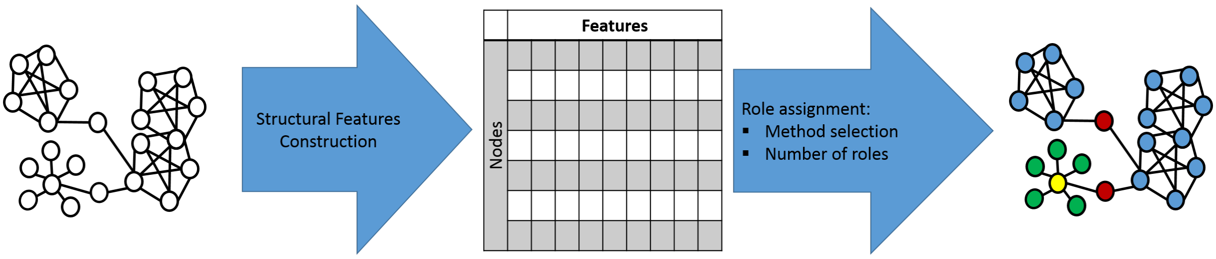


Figure 2.2: The overall framework of role mining with the input of network $G(V, E)$ and output of role membership or node V regarding the connections E. This framework involves two main steps: 1) construction of structural features; 2) role assignment.

2.2 Role mining framework

An overview of the general role mining framework can be seen in Figure 2.2. It is essentially constituted by two major steps. The first step involves transforming the graph representation into a feature representation. This is a crucial step and one should take great care to choose a set of structural features that can really describe and distinguish the nodes in the graph. The second step is the assignment of roles to nodes that are similar regarding their feature set. The next sections describe in more detail these two steps.

2.2.1 Structural features construction

A core element of the role mining process is constructing a feature vector that is a good representative of the fundamental structures in the graph, and capable of distinguishing node classes/roles. The feature vector may not be the same for different domains, such as biological networks or social networks, or for a specific application such as anomaly detection in computer networks.

In this thesis, our focus is on structural roles and thus only purely topological features are considered for extracting the roles. There is a wide range of structural features that measure different aspects of a node such as its degree, or its clustering coefficient. Some of these features may be very informative but expensive to compute, while others may be less informative but simpler and faster to compute. Generally speaking, there are two main groups of features that have been used for role mining: individual structural features and aggregated structural features. We explain each of these groups in the next subsections.

2.2.1.1 Individual structural features

Structural features are calculated for nodes based on their connections to the rest of the network, such as degree or clustering coefficient. These features are directly measured from network connectivity. Costa et al. gave a very comprehensive list of these features [CRTB07]. Structural features are categorized under different types such as distance-based, community-based, degree-based and etc. Many of these categorizations may overlap and one feature can be classified under more than one type. For example, closeness centrality can be considered as distance based feature or can be considered as a centrality measure. Here, we explain a number of most common categorization of structural features as follows:

- **Degree based features**: An important feature of a node in the network is its degree which is measured as the number of its direct connections to other nodes [DM04]. Many other features can be derived from the degree of a node. One of the most important is the *degree distribution*, $P(k)$, which shows the fraction of nodes in a network with degree k. The *correlations* between the degrees of different nodes is often of interest. The most basic approach to measure the correlation between nodes is to find the joint degree distribution $P(k, k')$.

- **Distance based features**: Distance is an important characteristic that depends on the overall network structure. The geodesic path between node u and v, is one of the paths connecting these nodes with minimum length. The length of the geodesic paths is the geodesic distance d_{uv} between nodes u and v. A number of features can be defined based on the nodes distances such as the *average distance* of node u defined as:

  $$l(u) = \frac{1}{N-1} \sum_{v \in V(G)} d_{uv} \tag{2.2}$$

  where N is number of nodes in graph G and $V(G)$ is the set of nodes of G. Another interesting feature that can be derived from distance is the *vulnerability* of a node. This feature finds critical components of a network by looking for the most vulnerable nodes and is defined as:

  $$V(u) = 1 - \frac{E_u}{E} \tag{2.3}$$

  $$E = \frac{1}{N(N-1)} \sum_{v \neq u} d_{uv} \tag{2.4}$$

  where E is the global efficiency of the original network and E_u is the global efficiency after the removing the node u and all its edges.

- **Centrality features**: Ranking of nodes in complex networks such as social networks is one of the research questions. Centrality measures assess the importance of a node in the whole network by assessing how much a node is central according to some criteria. This group of structural features has some overlap with other groups. For example, *degree centrality* ranks nodes based on their degree, hence it is a degree-based feature too. *Closeness centrality* and *betweenness centrality* are two other very common centrality features that are calculated based on the distance of a node to the other nodes in the network. Closeness centrality counts the average number of hops a node is away from the rest of the network and it is equal to average distance as measured by equation 2.2. Betweenness centrality is the the number of shortest paths a node u is part of them and is calculated as:

  $$b(u) = \sum_{v \neq u \neq y} \frac{\sigma_{vy}(u)}{\sigma_{vy}} \tag{2.5}$$

  where σ_{vy} is the number of paths between nodes v and y and $\sigma_{vy}(u)$ is number of those paths that pass through u. The fourth centrality feature that we introduce here is *eigenvector centrality*. This feature assesses the centrality of a node based on its connection to other central nodes. In other words, the centrality score of a node is higher if it is connected to high score nodes. This feature is measured by calculating the eigenvector of the network G based on the eigenvector equation $A\omega = \lambda\omega$, where A is the adjacency matrix of G, λ is eigenvalue and ω is the eigenvector.

- **Egonet based**: Ego is an individual node in the network, and an egonet is the subgraph of all individual nodes to whom ego has a connection of some path length. A node can have several s-size egonets containing nodes connected to the ego at distance s. Measurement of this group of features is restricted to the local neighborhood of the nodes. Some examples are:

  - Normalized node degree: quantifies the linkage of node u; it is the degree of node u divided by the sum of all nodes' degree in the network.
  - Normalized average degree: shows the intensity of connectivity in the neighborhood of node u; it is calculated by averaging over all degree of immediate neighbors of node u.
  - Standard deviation of degree: coefficient variation of the degrees of the immediate neighbors of a node. This feature characterizes the coherence of the connectivity; it is measured by the standard deviation of the degrees in the neighborhood of node u.

- Clustering coefficient: quantifies the connectivity between neighbors; it is measured as the proportion of existing connections between neighbors of node u to the number of all possible links between them [WS98].
- Locality index: characterizes the structure of neighbors' connectivity to the rest of the network; it is the ratio of links to the nodes outside of neighborhood to the number of links within the neighborhood to.
- Common neighbors : measure the commitment of nodes to the neighborhood. This feature shows if neighborhood of a nodes has an overlap with its neighbors. It is the number of common neighbors between a node's direct connections.

  $$CN(u) = \sum_{v \in \tau_u} \frac{|\tau_u \cap \tau_v|}{|\tau_u \cup \tau_v|} \tag{2.6}$$

  where τ_u is the set of neighbors of node u.
- Min-wise hashing: This is an established method to efficiently calculate the proportion of shared neighbors of a node, where the neighborhood of a node is defined by the set of nodes in its adjacency list [BCFM00]. The min-wise hash of such set can be generated by applying a permutation π and then taking the minimal value after the permutation. Let τ_v be the neighborhood of node v, then its min-wise hash value under π, $h_\pi(\tau_v)$ is:

  $$h_\pi(\tau_v) = \underset{u \in \tau_v}{min}(\pi(u)) \tag{2.7}$$

  where $\pi(u)$ is the value of u after permutation π. A min-wise hash signature of length k for v is generated by randomly drawing k permutations and concatenating the corresponding hash values. The same set of permutations are applied to all adjacency lists to generate the corresponding length-k signature for each node.

The egonet based features have the advantage of measuring the connectivity of a node in its neighborhood structure and it is also fast to compute. This type of structural feature is mostly used in the our proposed methodology in the following chapters.

2.2.1.2 Aggregated structural features

Aggregated features are constructed from structural features by applying an operation such as sum, maximum or average over the measured individual structural features, as explained in section 2.2.1.1.

Henderson et. al [HGL$^+$11] proposed a structural feature discovery algorithm in graphs that employs an exhaustive feature search strategy to extract local and egonet features. They find aggregated structural features for a node based on counts (weighted and unweighted) of the number links adjacent to a given node v and adjacent to the egonet of v. Their method also aggregates egonet-based features in a recursive fashion until no informative feature can be added. Examples of these recursive features include degree and number of within-egonet edges, as well as aggregated features such as "average neighbor degree" and "maximum neighbor degree".

Alternatively, we may construct features using a guided strategy [DBdCD$^+$05, MHC$^+$08] where a heuristic is used to identify relevant features. We can evaluate the relevance of features by using different measures such as Pearson correlation (and Spearman rank correlation) [Cha07], information gain [Yao03], or Bayesian information criterion (BIC) [HQ79, S$^+$78]. These are basic methods for unsupervised features selection where the main goal is to select features with high distinguishing power and avoiding redundancy in the feature set. Assumptions and knowledge about the application domain can also be used as further guidance for feature selection. For example, the classification error can be used as a measure for evaluating the features set if there is a class label for nodes in the network.

In this class of feature search strategies, Rossi and Ahmed showed a search algorithm over a space of features such as degree, egonet-features and other variants. In each iteration they use a set of recursive relational operators (e.g., sum, mode, etc.) to build new features. The new features are evaluated regarding an evaluation metric, and this process repeats until there are no more novel/useful features being generated.

Both of these feature construction strategies show a drawback in which the generated roles are typically more difficult to interpret. However, they are very comprehensive and can capture arbitrary structural patterns.

2.2.2 Role assignment

The second step in a role mining framework is to decide on how to assign nodes with similar feature vectors to the same role. Role assignment can be seen as a partitioning problem where we group the nodes into different classes. The two main classes of algorithms for this task are clustering methods and low-rank approximation techniques. One of the main challenges in both methodologies is selecting the best number of roles in a network which will be discussed in section 2.2.3. In this section we review the state of the art in the usage of these methods for role mining and we discuss other possible approaches.

2.2.2.1 Clustering algorithms

As the role assignment setting resembles to a clustering problem, any clustering algorithm can be used for this purpose. There are essentially two types of algorithms used for role assignment: partitioning algorithms such as k-means [Ber06, Zhu05] and hierarchical clustering algorithms such as agglomerative or divisive clustering [MC12]. Most of the clustering methods such as k-means are hard-clustering techniques, in contrast to soft clustering methods which allow nodes to be in multiple clusters. A few classical methods are fuzzy C-means [BEF84] or types of Gaussian Mixture Models [Ras99], among others [EF05]. In this section, we review some the the clustering algorithms that will be utilized in the following chapters of this thesis.

K-means: is a data partitioning algorithm, which divides data into several subsets. The k-means algorithm [HW79] is by far the most popular clustering tool used in scientific and industrial applications. In this algorithm, each cluster C is represented by the mean (or weighted average) of its points, the so-called centroid. The sum of discrepancies between a point and its centroid, expressed through an appropriate distance, is used as the objective function. For example, the L_2-norm based objective function, the sum of the squares of errors between the points and the corresponding centroids is equal to the total intra-cluster variance

$$E(C) = \sum_{j=1}^{K} \sum_{x_i \in C_j} \|x_i - c_j\|^2 \tag{2.8}$$

In principle, the optimal partition, based on the objective function 2.8, can be found by enumerating all possibilities. But this brute force method is infeasible in practice, due to the expensive computation involved. Therefore, heuristic algorithms have been developed in order to seek approximate solutions. One of the most utilized methods is iterative optimization known as Forgy's algorithm [For65]. It consists of two-step major iterations that (1) reassigns all the points to their nearest centroids, and (2) recomputes centroids of newly assembled groups. Iterations continue until a stopping criterion is achieved (for example, no reassignments happen).

Hierarchical clustering: organizes data into a hierarchical structure according to the similarity matrix between data [Joh67]. The results of hierarchical clustering (HC) are usually depicted by a binary tree or dendrogram. The root node of the dendrogram represents the whole data set and each leaf node is regarded as a data object. The intermediate nodes, thus, describe the extent that the objects are proximal to each other; and the height of the dendrogram usually expresses the distance between each pair of objects or clusters, or an object and a cluster. The ultimate clustering results can be obtained by cutting the dendrogram at different levels. This representation provides very informative descriptions and visualization for the potential data clustering structures, especially when real hierarchical relations exist in the data, like the data from evolutionary research on different species of organizms. HC algorithms are mainly classified as agglomerative methods and divisive methods. Agglomerative clustering starts with clusters and each of them includes exactly one object. A series of merge operations are then followed out that finally lead all objects to the same group. Divisive clustering proceeds in an opposite way. In the beginning, the entire data set belongs to a cluster and a procedure successively divides it until all clusters are singleton clusters.

Spectral clustering : is usually used for graph partitioning problems where a graph-based measure is to be minimized. An example is the normalized cut [VL07] algorithm, which clusters objects based on the eigenvectors of their similarity matrix. For the nodes and their similarity, the graph Laplacian L is built: $L = S - W$ where S is the degree diagonal matrix of similarity graph of nodes, W is the similarity matrix of data. Then the first k eigenvectors of L are calculated. Finally the clustering is derived by applying k-means on a matrix, built from concatenation of the first k eigenvectors as columns.

Ensemble clustering: can improve accuracy of results by aggregating multiple partitionings to alleviate the noise [TLJF04, FJ02, GMT05, SG03]. It can be used in different applications such as network community discovery [AUP07] or monitoring of communities evolution[LF12]. Given K partitionings over a set of objects, the objective of ensemble clustering problem is to obtain a single aggregated clustering. The ensemble clustering λ is the one that best matches with every base clustering. In other words, λ must minimize the cost function $\sum_{i=1}^{K} Dist(\lambda, C_i)$. In this problem setting, the distance between clusterings is measured only by the cluster label and without accessing the original feature space of data.

Strehl and Ghosh measured the cost function in terms of shared information between clusterings [SG03]. They used normalized mutual information (NMI) to measure the similarity of clusterings. Since finding the optimal combined clustering over the defined cost function is computationally expensive, they used heuristic solutions instead of optimization. Gionis et al. defined the cost function as the number of mismatches between the clusterings [GMT05]. They proposed a number of approximate algorithms to find the aggregated clustering. The common approach of all the proposed methods is to build a new similarity matrix between the objects to be clustered, using the clustering co-occurrence instead of their original feature space. This similarity matrix is used either directly to re-cluster the objects or to build a graph similarity of data and then derive the clustering by partitioning the graph.

Evolutionary clustering: is defined by Chakrabarti et al. as *“the problem of processing time-stamped data to produce a sequence of clusterings; that is, a clustering for each time step of the system. Each clustering in the sequence should be similar to the clustering at the previous time step, and should accurately reflect the data arriving during that time step"* [CKT06]. Evolutionary clustering finds application in the domains where the properties of objects change over time due to concept drift or noise. In such problem, at each time step a new set of data arrives to be clustered. To cluster the new data one needs to observe its structure: 1) if the structure of the new data does not change significantly, or the changes are due to noise, then it is clustered as in previous time; 2) otherwise, the clustering must be modified in a way to reflect the actual structure of new data and detect the deviations. These two objectives are modeled as cost functions in evolutionary clustering, called temporal cost (TC) and snapshot cost (SC) respectively. The overall cost of clustering at current time t is defined as follows:

$$cost = \alpha * SC(C_t, X_t) + (1 - \alpha) * TC(C_t, X_{t-1}) \tag{2.9}$$

where C_t is the clustering of data X_t at time t and α is a user defined parameter to adjust the importance of historical data. Chakrabarti et al. modified hierarchical and k-means clustering algorithms to incorporate the defined cost function [CKT06]. They measure the distance between the clusters across time by pairing the centroids of clusters. Another pioneering work in this area is by Chi et al. [CSZ$^+$07]. They proposed two frameworks for evolutionary clustering, the first one assesses the temporal cost at the data level, meaning it evaluates the new clustering on the old data. The second one does the evaluation at model level, comparing the clusterings with each other using Chi-square statistics. They incorporate the cost functions into a spectral clustering framework and solve its relaxed version to derive the partitioning of the data.

2.2.2.2 Low-rank approximation

Low-rank approximation methods are another group of methods one can use for role assignment. These methods find K roles from a large feature matrix X by computing a low rank-K matrix $\hat{X}$ that best approximates the original feature matrix with respect to any standard matrix norm. There are many possible dimensionality reduction methods suited for this purpose. Some examples commonly used are Singular Value Decomposition (SVD) [GR70], Principal Component Analysis (PCA) [Jol05] or Non-negative Matrix Factorization (NMF) [WZ13].

An example of a low-rank approximation method in the context of a structural role mining problem is given by Henderson et al [HGER^{+}12], where they used non-negative matrix factorization for role assignment. In this method, a rank K approximation with two matrices W and H is generated for the feature matrix X, such that $WH \approx X$ where each row of $W \in \mathbb{R}^{N \times K}$ represents a node's membership in each role, and each column of $H \in \mathbb{R}^{K \times M}$ represents how membership of a specific role contributes to estimated feature values. More formally, given a nonnegative matrix $X \in \mathbb{R}^{N \times M}$ and a positive integer $K < min(N; M)$, the goal is to find nonnegative matrices $W \in \mathbb{R}^{N \times K}$ and $H \in \mathbb{R}^{K \times M}$ that minimize the function $f(W, H) = \frac{1}{2}||X - WH||^2$.

All of the low-rank approximation methods need three elements: (i) a similarity/objective function (e.g., Frobenius norm, KL-divergence); (ii) regularization terms if warranted (e.g., sparsity constraints, L2, etc); (iii) a solver (e.g., Multiplicative update). Henderson et al. [HGER^{+}12] used NMF-based approach for roles assignment. They modeled the objective function via KL-divergence with L2 regularization and used Multiplicative update as the solver. One may also add sparsity and other constraints [HS06, CPC08] to these approaches to better adapt the roles for specific applications [GERD13]. We also note that many of these techniques may also be used for learning roles over a time series of graphs [RNGH13].

There are however some issues on the application of these basic methods to the role mining problem, as we show in section 2.4, as it the case, for example, in the discovery of roles in dynamic time evolving networks. Therefore, there is a need for novel methodologies capable of tackling these issues. We cover dynamic role mining in Chapters 3 and 5

2.2.3 Number of structural roles

One of the challenges in role mining is to determine the appropriate number of roles for every method explained in the previous sections. This resembles the task of finding the number of clusters for a clustering algorithm which has received a substantial amount of attention in the literature. Nevertheless, this stills remains an open question for a general case. Some of these methods are based on heuristics, while others have a more fundamental basis in statistics (e.g., Akaike information criterion (AIC) [BMA83]) and information theory (e.g., Minimum Description Length, known as MDL) [Grü07].

A general approach is to define a cost function and gradually increase the number of roles as long as the cost of the role model decreases (or likelihood improves using that number of roles). For example, using MDL to automatically determine the number of structural roles is intuitive to learn roles such that the model complexity (number of bits) and model errors are balanced. Note that deriving a large number of roles increases model the complexity, but at the same time decreases the amount of errors. In contrast, using less roles decreases model complexity, but increases the amount of errors. The cost function for selecting the number of roles can be the cross validation error, or f-score, if a class label exists in the domain application. In this case, the number of roles is iteratively changed until the best classification error is achieved.

2.3 Applications

In this section, we discusse the application of role mining to many network analysis problems.

User classification: the role membership of nodes in a network is a good representative to be used in a user classification task in a network [MGA07, LG14].

For example, in a IP-to-IP network, we may want to infer the classes of traffic (e.g.,Web, DNS, SMTP, P2P) [MHC$^+$08]. Nodes in each class have different structural roles that can distinguish them. Rossi et al. [RGNH12] showed that the IP addresses follow different temporal structural behavior which represent their roles in the netowrk. Zao et al. [ZWY$^+$13] used structural role to infer social status of users in the Linked network in IT industry. There are four social roles : Research & Development (R&D), Marketing & Sales (M&S), Human Resource (HR) and Executives (EXE). Generally, the classification task is to find label of nodes in a network by using the structural role membership of matrix as the predictors.

Network comparison: Another interesting application of role mining is in the domain where two networks G and H are compared based on their role profiles [RFT13]. In network comparison, the goal is to provide a meaningful measure of structural similarity for a given set of networks. For example, comparison of co-authorship graphs across different scientific disciplines can reveal the different collaboration patterns in each field [CRBS12]. Berlingerio [BKERF12] proposed a framework for network comparison based on structural features. In this framework, structural roles of networks are used as a *signature* vector and the similarity score of every pair of networks is measured as the distance between their set of structural roles.

Anomaly detection: The results of role mining can be beneficial to anomaly detection by finding anomalous nodes (or links) with unusual role memberships (static graph-based anomaly) or nodes with unusual role transitions (dynamic graph-based anomaly) [HGER$^+$12, RNGH13].

2.4 Discussions

We examine additional aspects of role mining in this section, including role mining in dynamic networks and role mining in communities.

2.4.1 Dynamic networks

One of the main aspects of networks is their dynamics and it is important to reflect this feature in role mining. For example, in a social network, users' preferences and behavior is time-dependent and the roles of users are tied to their historical connections. Hence, the process of role mining in a dynamic network must extract roles that not only reflect the current structure of network but also reflect the historical behavior of nodes. This is still a relatively new research area with a scarce amount of work done on it. Rossi et al. [RGNH12] used roles for dynamic network analysis, discovering patterns in dynamic networks, and for predicting future structural transitions in those networks. They extract dynamic roles by applying NMF on local primitives (i.e., degree/egonet-based features). Their method learns features from the time series of graph data, then assigns roles using those features. Using the learned feature and role definitions, they now extract feature-based roles in a streaming fashion for detecting graph-based anomalies. With exception of the aforementioned paper, we are among the first to tackle this issue [CSR11, CRS12a, CSRPar]. Incorporating the dynamics of networks in the role mining is at the heart of this thesis and we have proposed a series of methods to first examine the dynamics of roles (Chapter 3) and then we incorporated the temporal behaviors into a novel role mining method (Chapter 5).

2.4.2 Role in communities

Community detection is an essential task in the field of network analytics, and it has received extensive research interest [TBWK07, AY05, GN02, NG04, LLM10]. Community detection aims to identify groups of nodes that are densely connected between themselves, when compared with their neighbors. These methods find applications in several domains such as finding clusters of users from social networks and functional protein complexes from bioinformatics networks. In real world networks, nodes may play different roles in different communities. For example, nodes that interface with other communities and nodes that are peripheral to community cores, and star nodes that acts as bridges when connecting multiple tight knit communities. Ruan and Parthasarathy [RP14], proposed a method that simultaneously does the role and community assignment. They constructed structural feature vector for nodes using min-wise hashing [BCFM00] and iteratively assigned nodes to roles by incorporating the community membership information. They showed how the role assignment of a node also depends on the communities it belongs to.

2.4.3 Evaluation metrics

One of the challenges in role mining is how to evaluate the quality of discovered roles, and to evaluate the capability of various methods. The fact is that, in the role mining problem there is no a-priori class to be used as ground truth. Roles are used as a descriptive modeling tool to understand the "roles" played by actors in the network and the nature of the methodology is unsupervised. The principle evaluation method in previous work is to use roles for an application (e.g., link prediction, anomaly detection). An evaluation strategy is to employ domain application knowledge to guide a role learning method. For example, if the end goal is to discover roles for node classification, then classification error is a good measure to examine performance of the role mining method. In this thesis we followed the same strategy to demonstrate the performance of our methods. We demonstrate the efficacy of our methods by using domain application knowledge.

Chapter 3

Structural role mining and tracking

How are structural roles formed and changed over time? When do significant changes occur in the role configuration?

In this chapter, we study the dynamics of complex networks by examining the emergence of structural roles and their evolution. This gives a good understanding of the characterization of network dynamics by tracking the evolution of roles of nodes over time. We introduce two methods to discover the evolution rules that describe how, with time, a role changes to another. The first method is a more general methodology where by defining time granularity of evolution, a set of rules are extracted to describe the transition of roles from one to another. In the second one, we define a series of events and find the transition interval for every specific event. Both methods examine the dynamics of roles but with different approaches. The first one has the advantage of being more general and being capable of finding all evolution rules with different time granularity and the second is more exact and objective in that it only searches for pre-defined events and detects the relative transitions.

For a given evolving network $G_t = \cup_{i=1}^{t} G_i$,we are interested in examining how the role configuration of nodes at each time step is and how it changes. A dynamic network of $G_t = (V_t, E_t)$ consists of $V_t = \cup_{i=1}^{t} V_i$ which is the set of unlabeled nodes at time t and E_t the set of connections between V_t. In the whole lifetime of G, the nodes are constant but the edges may appear and disappear. More formally, our goal in this chapter is to find a set of roles configurations $\{L_1, ..., L_t\}$ for nodes in the network at time $i = 1$ to t in order to find a set of rules ϕ such that it represents the dynamics of roles, in particular how they change from one to another. Hence, we first find L_t for G_t using a static role mining method, explained in section 3.1, then we find ϕ from $\{L_1, ..., L_t\}$ using an association rule mining based method, and an event detection method. The two methods are explained in sections 3.2 and 3.3.

3.1 Static structural role mining

For static role mining we could follow any of role assignment methods explained in section 2.2.2. As this step is not our main focus in this chapter, we design a simple method to find roles of nodes in the network for each time step. In our method, the label of a node is automatically determined based on its properties in the network using the k-means clustering algorithm. The same role label is assigned to the nodes that are in a similar position and have similar properties. We assess the distance of two nodes by their properties, rather than using the number of edges between them. Two nodes are close if they have a similar *feature vector*.

3.1.1 Structural features

The first step in structural role mining is to select a set of local measurements that best characterize nodes in the network structure. Feature construction for structural role mining, like any other machine learning problem, follows the feature selection and construction process where a subset of features are selected from the original features so that the feature space is optimally reduced according to a certain criterion. In this process, a set of new features may also be created and used either in isolation or in combination with the original features. The main goal of this step is to improve performance measured in terms of estimated accuracy, visualization and comprehensibility of learned knowledge. Following this general process, in this thesis, we selected the structural properties of nodes that have been shown to be correlated to social roles of users [CRHK09].

Out of different categories of structural features discussed in chapter 2.2.1, we mostly use individual structural features as they are easier to interpret. Our focus is more on studying local neighborhood of nodes, hence we use the subset of egonet features explained in section 2.2.1.1, including:

- Normalized node Degree (ND)
- Normalized Average Degree (NAD)

- Standard Deviation of Degree (SDD)
- Clustering Coefficient (CC)
- LOCality index (LOC)

This feature vector has the advantage of measuring the connectivity of a node in the neighborhood structure and being relatively scalable as it only considers the neighborhood of a node for calculation.

3.1.2 Number of roles

Role mining is an unsupervised methodology and the number of potential roles in the network is not known in advance. This task is equivalent to determining the number of groups in a dataset which is a fundamental and largely unsolved problem in cluster analysis. In this chapter, we employ the *"Jump method"* explained in [SJ03], since it does not require parametric assumptions, is independent of the clustering algorithm, and was shown to achieve excellent results. This method uses a theoretic information approach that considers the transformed distortion curve $d_K^{-p/2}$ [SJ03]. *"Distortion"* is a measure of within cluster dispersion which is the Euclidean distance between the data and the set of cluster centers as a function of the number of clusters, K.

First, this method runs the clustering algorithm for different numbers of clusters, K, and calculates the corresponding distortions, $\hat{d}_K$. In this thesis we use k-means algorithm for clustering and vary the value of K from 2 to $N/2$ where N is the number of nodes in the network. Then it transforms the distortion by power transformation of $y = p/2$, where p is the number of dimensions in the dataset. The "jumps" in the transformed distortion are calculated by $J_k = \hat{d}_K^{-y} - \hat{d}_{K-1}^{-y}$. Finally, the appropriate number of clusters for the data is equal to $K^* = argmax_k(J_k)$.

3.1.3 Role assignment method

We use multivariate statistics and pattern recognition techniques [JW02] to find groups of identical nodes as structural roles in a network. Clustering is a method widely used for finding groups of objects in the dataset, called clusters, such that the objects in the same group are more similar to each other than they are to objects of other groups. We use the well known k-means clustering algorithm [HW79], which bases its operation on the euclidean distance between nodes. This distance is calculated for every two nodes by considering all five features. We use the normalised feature vectors into interval [0,1] of nodes to calculate their pairwise distances, for two nodes u and v the distance is :

$$distance(x_u, x_v) = \sqrt{\sum_{i=i}^{M} (x_u^i - x_v^i)^2} \tag{3.1}$$

where x_u is the feature vector of node u, x_u^i is the ith feature of u and M is the number of features. Each cluster contains nodes with a similar position in the network regarding their feature vectors. Hence, the same role or label can be assigned to them.

We use the Jump method to determine the number of groups of nodes. We do the clustering on the aggregated dataset that includes feature vectors of every node for the whole lifetime of the network. At the end of this phase, the coherent groups of nodes are derived and labeled. Therefore, a sequence of labels is generated for each node over time, that determines to which cluster a node belongs at each time. In the next sections, we introduce two methods to examine the dynamics of the derived roles considering these sequences.

3.2 Node role evolution patterns

After finding roles of nodes in each time step by the method explained in section 3.1, we introduce our first method for studying dynamics of roles in this section. A first insight of the roles set over time shows that roles are not constant (Figures 3.2a). Some new roles emerge over time and some disappear. Role mining over an evolving network results in a set of roles configurations $\{L_1, ..., L_T\}$. In other formulation for each node u, there is a vector r_u that includes the sequence of roles $r_u[t]$ representing the role of u at time t. To extract the patterns of changes, we adopt an association rule mining framework [AIS93]. Association rule mining is a popular method for discovering relations between variables.

Agrawal et al. [AIS93] originally defined the association rule mining problem as follows: For a set of n binary variables, called items, $I = \{i_1, i_2, \ldots, i_n\}$ and a set of transactions $\Gamma = \{\gamma_1, \gamma_2, \ldots, \gamma_m\}$, a rule is defined as an implication of the form $A \Rightarrow B$ where $A, B \subseteq I$ and $A \cap B = \emptyset$. Each transaction in Γ contains a subset of the items in I and A, B are called itemsets.

In our problem setting, we are interested to find rules of the form $(r, t) \Rightarrow (r', t')$ that explain role r at time t transformed to role r' at time t' . Hence, we consider the sequences of roles for users $\{r_u | \forall u \in V\}$ as the transactions where the items are roles at time steps $[1, T]$. This results in very large transaction size of $T \times K$ where K is the number of roles discovered in the evolving network. We use a sliding window method to investigate the evolution rules in each time segment $w = [t_i, t_j]$. This facilitates rule extraction and helps scalability of methods as it reduces the number of items. In order to find interesting rules out of all possible rules, some constraints are used over the rules: threshold on support and confidence. Support assesses how often a rule applies to the given dataset and confidence measures how frequently items in B appear in transactions containing A, defined formally as:

$$\begin{aligned} \mathrm{supp}(A) &= \frac{\sigma(A)}{|\Gamma|} \\ \mathrm{conf}(A \Rightarrow B) &= \frac{\mathrm{supp}(A \cup B)}{\mathrm{supp}(A)} \end{aligned} \tag{3.2}$$

where $\sigma(A)$ is the number of transactions in the data set that contain A, and $|\Gamma|$ is the total number of transactions. Frequent itemsets are the sequences or subsequences that have minimum support. Therefore, the patterns of node evolution are the extracted frequent itemsets and association rules.

We exploit the Apriori algorithm to find the evolution pattern of roles. Apriori algorithm is a powerful tool for mining associations, correlations, causality and sequential patterns [AS94, AS95, BMS97, KMR$^+$94]. Association rules mining has two main steps [AS94]:

1. Finding all sets of items (itemsets) whose transaction support is above a minimum support threshold. The support for an itemset is the number of transactions that contain the itemset. Itemsets with minimum support are called large itemsets, and all others are called small itemsets.

2. Use the large itemsets to generate the desired rules for every large itemset γ and all non-empty subsets of γ. For every such subset γ_j, output a rule of the form $\gamma_j \Rightarrow (\gamma - \gamma_j)$ if the ratio of support (γ) to support (γ_j) is at least the minimum confidence. We need to consider all subsets of γ to generate rules with multiple consequents.

The Apriori algorithm generates the frequent itemsets with different time granularity. Patterns of evolutions are generated using a sliding window method that enables us to detect changes at different stages of the network lifetime. At each time window, rules with different time granularity are extracted.

3.2.1 Data

We evaluated our evolutionary rule detection method on two different datasets, a network of the world countries' global trade (GDP data) [Gle02], and a synthetic scale free network. We start by describing these datasets in some detail and then present our evaluation results.

The first data set is created from the publicly available Expanded Trade and GDP Data [Gle02]. The data represents the yearly imports and exports, total trade and gross domestic product (GDP) of 196 countries spanning for 52 years from 1948 till 2000. The time series for each country is the proportion of its share in the global economy according to its GDP for that year. The time series for GDP-Norm is the normalized value of each individual annual GDP, divided by the total GDP for all countries during that year. The topology for the graph was created by comparing the yearly total trade for each country and its trade with each of the other countries. If the trade between country A and country B in any given year accounts for more than 10% of either country's total trade for that year, an edge is created between the two countries.

The second dataset is a synthetic scale-free network generated based on the Barabasi-Albert model for graph generation [BA99]. It is a model of network growth that is based on two basic parameters: growth and preferential attachment. The basic idea is that in the network nodes with high degrees acquire new edges at higher rates than low-degree nodes. An undirected graph is constructed as follows. Starting with m_0 isolated nodes, at each time step $t = 1, 2, \ldots, N$ a new node j with $m \leq m_0$ links is added to the network. The probability that a link will connect j to an existing node i is linearly proportional to the actual degree of node i given by

$$P(k_i) = k_i / \sum_j k_j \tag{3.3}$$

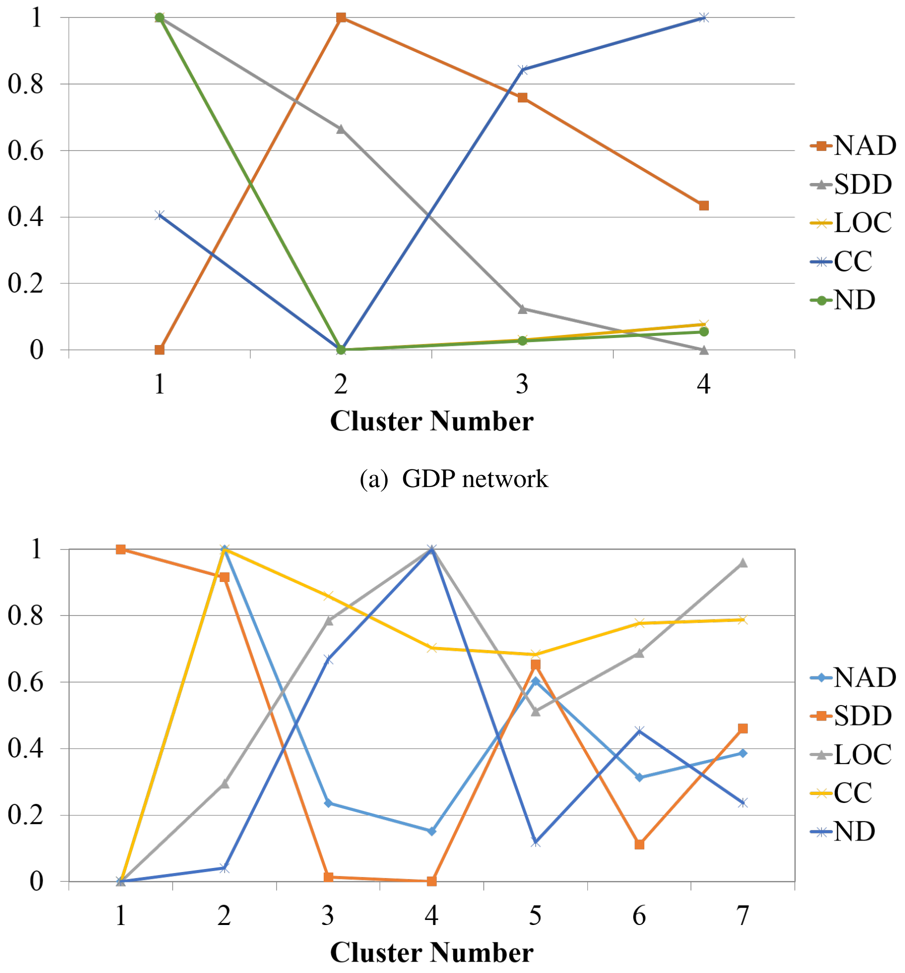
3.2.2 Experimental results

Table 3.1 provides details on the networks used in our experiments, namely the number of nodes in each network, the number of time instances of network evolution, and the number of roles discovered by static role mining method explained in section 3.1.

Figures 3.1a and 3.1b show the profile of the structural roles in each network. The profile depicts the values of the average feature vector of each role in the network. As explained earlier, the feature vector includes the metrics normalized average degree (`NAD`), coefficient

Dataset	# nodes	# times	# structural roles
GDP	171	52	4
Scale-Free	200	100	7

Table 3.1: Statistics of used networks



(a) GDP network

(b) Scale-Free network.

Figure 3.1: Profile of roles in the GDP and Scale-Free network.

variation of the degrees of immediate neighbors (SDD), the clustering coefficient (CC), the locality index (LOC), and the normalized node degree (ND).

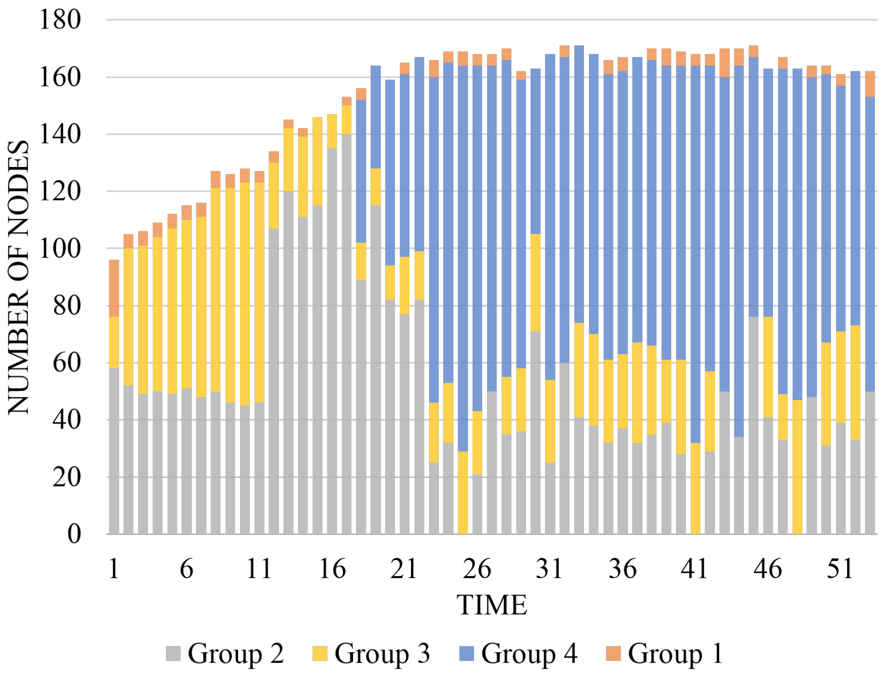
GDP network: in this global trade network of countries, our method found four distinguishable structural roles. Each role has a different feature vector as illustrated in Figure 3.1a. The first role includes nodes that represent countries with very high degree and many low degree nodes connected to them. Neighbors of these nodes have low degree since the normalized average degree of the immediate neighbors of a node for this group is very low. This means that nodes of role one behave as hubs in the network, that is, as hub countries in global trade, with commercial transactions with many other countries that have a high variation of degree in neighborhood (SDD). According to the value of LOC and CC, respectively, the locality index and the clustering coefficient, nodes of this group are highly connected in their neighborhood. Examples of countries with this role are United States of America, Canada and France.

Figure 3.2a depicts the evolution of the frequency of each role over time in each network. At the initial stages of network evolution, roles number one and three are rather common, but they become rare as the network evolves. These roles have different sizes (number of nodes) at each time step, but they never vanish. Over time, one can notice a transition from role number three to role number two. After the initial stages of network lifetime, a new group emerges, in this case role four.

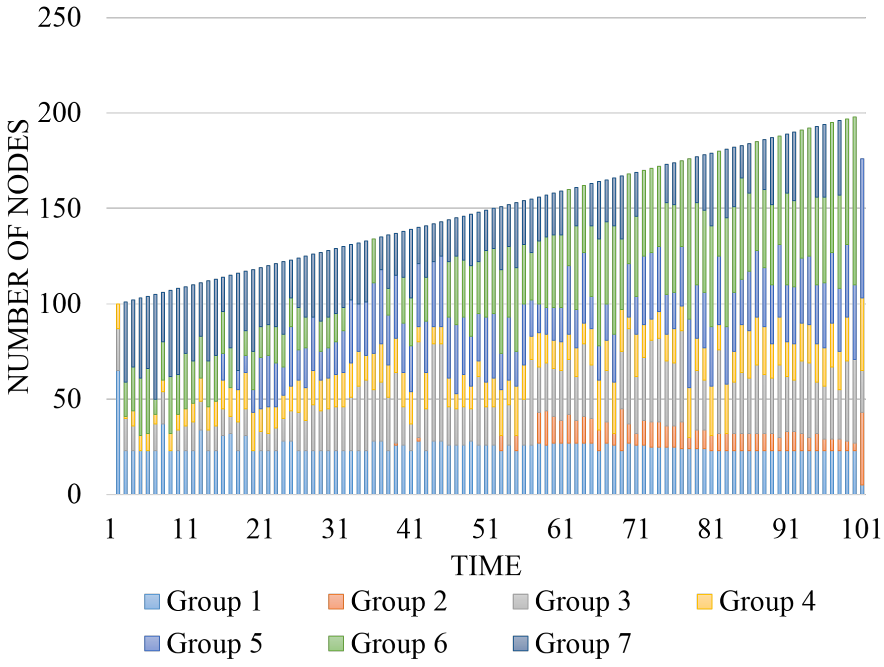
We find the evolution of roles over time by extracting association rules using the method described in section 3.2. Table 3.2 shows the strongest rules for the datasets, in terms of support and level of confidence. We used a sliding window to find out the changes in the network. The sliding window parameter helps to narrow down the search interval to find more precisely rules that describe the dynamic of roles. The feasible sizes for the window can be determined by observing Figure 3.2a, which shows the trend of node's membership. With different sliding window sizes, several rules could be found. The most significant ones that characterize the appearance and disappearance of the groups are listed in table 3.2.

For example role number four does not exist in the network before time step 18. This pattern of change is detected and described by the rule $\{t14 = 2, t17 = 2\} \Rightarrow \{t18 = 4\}$. This rule says that nodes with role two at time 14 and 17 are likely to change their role to four at time 18. The support for this rule is 11%, but its confidence is 87%.

Scale-Free network: this network was generated with 200 nodes and we sampled 100 networks from its evolution time. Our method detected seven different roles with distinct



(a) GDP network



(b) Scale-Free network

Figure 3.2: Frequency of roles in the networks over time.

Network	Rule	Support	Confidence
GDP	$\{t1 = 2, t4 = 2, t8 = 3\} \Rightarrow \{t9 = 2\}$	18%	95%
	$\{t14 = 2, t17 = 2\} \Rightarrow \{t18 = 4\}$	11%	87%
	$\{t28 = 4, t29 = 1\} \Rightarrow \{t30 = 3\}$	16%	70%
	$\{t40 = 2, t41 = 3, t42 = 3\} \Rightarrow \{t43 = 2\}$	17%	75%
Scale Free	$\{t16 = 7, t18 = 7\} \Rightarrow \{t20 = 5\}$	6%	72%
	$\{t57 = 5\} \Rightarrow \{t60 = 2\}$	11%	82%
	$\{t75 = 6, t76 = 6\} \Rightarrow \{t80 = 3\}$	7%	81%

Table 3.2: Derived rules for the networks

feature vectors, as illustrated in Figure 3.1b.

The first role in this network includes nodes that are weakly connected such that all of their local connectivity properties in the feature vector of this group have the lowest values between the nodes but have a very high variation of degree in their neighborhood (SDD). A reason for this is that the neighbors of these nodes are mostly low degree nodes that, however, are connected to a hub in the network with very high degree. As shown in Figure 3.2b, these roles have different size in each time but never cut down in the network lifetime. Second role includes nodes with low degree. This role was formed almost at the middle of network evolution time span (low ND). The nodes with this role are connected to very high degree nodes (high NAD and high SDD). This role appears after the 50th time instance. The third and forth roles include highly connected nodes (high ND and CC) with neighborhood of low degree nodes (low NAD and SDD). The other three roles, 5, 6 and 7, are low degree nodes, but the nodes with fifth role are also connected to a hub, which does not happen in the other roles. The sixth role emerges at the beginning stage of the network development and becomes more frequent as time goes by. However, role 7 emerges at initial stages of the network lifetime and its size remains almost constant over time.

Extracted rules in table 3.2 describe the strongest trends in nodes' transitions between groups. For example $\{t57 = 5\} \Rightarrow \{t60 = 2\}$ shows that nodes with fifth role, after a while, change their role to the second role. This rule also shows that as time goes on, regarding the generation model of the scale-free network, although the neighborhood of the nodes get more crowded (NAD and SDD increases), their degree remains low. Nodes in the fifth role have low degree, thus they can not absorb new connections and their degree does not increase.

3.3 Role event detection

In this section, we explain our second method for studying dynamics of roles in a network. Changes in a network are due to some basic events: node addition or deletion, and new edge addition or deletion. These events generate more complex behaviors in networks such as role formation or dissolution. We define five basic types of events for structural roles according to the changes in their size, that is, the number of constituent nodes. If the size of a role changes considerably, it shows that properties of a number of nodes has changed and subsequently the structure of the network is altered. In a certain time interval of the network evolution, a role can grow, shrink, emerge, dissolve or remain constant. Regarding these behaviors, we defined five events in a network life time as follows:

- **Growth**: A role grows if its size has a constant increasing trend in a time interval.
- **Shrink**: A role shrinks if its size has a constant decreasing trend in a time interval.
- **Emerge**: A new role emerges if its size has a constant increasing trend in a time interval and it does not exist in the previous intervals.
- **Dissolve**: A new role dissolves if its size has a constant decreasing trend in a time interval and it does not exist in the next intervals.
- **Constant**: A new role remains constant if its size does not change. In this case, there might be some nodes leaving or joining although the size of the role does not change considerably.

There are two primary event categories occurring in the network: shrink and growth. The others are specific cases of the original ones. For example, dissolve is a special case of shrink event where the size of the role shrinks to zero. All the occurring events in a network of any of these types are discovered and described by a two step method. In the first step, all time intervals where an event occurs are detected. We call such intervals as *transition intervals*. In the second step, a set of rules describing the events are generated which we call *transition rules*. Our goal is to discover rules that describe each event, for example we are interested in rules that can explain the origin of new nodes that have joined a role in a certain time interval.

3.3.1 Transition intervals

A transition interval is a time interval where a considerable number of nodes leave or join a role. For a given role r, the size of the role over time constitutes a time series denoted as $s_r(t), t \in [1, T]$. A transition interval is the subsequence of s_r which holds a constant increasing or decreasing trend. Hence, we extract transition intervals for s_r by segmentation of the time series. Starting from $t = 1$, $s_r(t)$ is approximated by linear regression to find the transition intervals. If the error of the fitted line for a subsequence $s_r[a : b]$ exceeds the threshold, the interval $[a, b]$ breaks to the point j where it gives the best approximation for $s_r[a, j - 1], j < b$. The error is measured in terms of the sum square of residuals. The threshold is controlled by the maximum number of arbitrary transition intervals. The maximum error is increased until the number of intervals is not more than the defined maximum number of intervals. For this method we can either define the maximum error for the linear regression, or the maximum number of desired intervals. The slope of the fitted line for each segment shows if the interval is increasing or decreasing which respectively determines the growth (emergence) or shrink (dissolve) events.

3.3.2 Transition rules

Having a list of transition intervals for a structural role, we extract a set of rules to describe how an event happened. A transition rule is of the form $r \rightarrow l$ shows that nodes from role r moved to role l. The order of transitions is important, since it shows the trend of changes in the network properties. For example, for the sequence $\{1, 1, 1, 2, 2, 2, 3\}$, the transition rules are $1 \rightarrow 2$ and $2 \rightarrow 3$. We extract these one-step transition rules for a transition interval by building a transition matrix in all transition intervals. The support count of a rule $r \rightarrow l$ is defined as the number of nodes that go from cluster r to l in that interval.

In the next section we apply this methodology on three different networks to characterize the dynamics of these networks and evaluate the applicability of our method.

3.3.3 Data

For our experiments, we used three different real complex networks: the global trade network of countries in the world, GDP data [Gle02] as explained in section 3.2.1, a network of USA airports[1] and a co-authorship network obtained from DBLP data [BBBG09]. We

[1] http://www.routeviews.org

Dataset	Time snapshots	$\|V\|$	$\|E\|$	Node growth rate	Edge growth rate
GDP	53	186	8839	2.47	7.93
USA airports	244	1919	14391	1.64	1.21
DBLP	11	31592	49599	3.4	4.57

Table 3.3: Datasets statistics: number of time-snapshots, number of nodes $|V|$ and edges $|E|$ at the final snapshot and node and edge growth rate (ratio between the final and initial time-snapshots).

use the undirected form of these networks. Table 3.3 overviews some topological features of the three studied networks.

USA airports: This data is the complete network of US airport from 1990 until April of 2011. We constructed monthly networks where two airports are connected if a flight was scheduled between them in that month.

DBLP: This is a co-authorship network from the DBLP data with a yearly time granularity. The nodes are authors that are connected in a certain year if they are co-authors in that year. It includes co-authorship data from 1992 to 2002 [BBBG09].

3.3.4 Experimental results

Here, we present the results of applying the second method, explained in section 3.3 on the datasets. Table 3.4 provides a brief overview of the derived results for the networks, namely the number of structural roles, the number of events and the number of rules, extracted in each network.

The methodology explained in section 3.3 gives us a profile for the network, including information about:

- The set of existing structural roles in the networks;
- The set of events occurring in the network in a time span.

Figure 3.3 shows the profile of the roles found in each network. The profile depicts the

Dataset	#roles	#events	# rules
GDP	4	22	66
USA airports	6	52	296
DBLP	8	25	195

Table 3.4: Datasets statistics: Number of structural roles found, numbers of detected events and number of extracted rules.

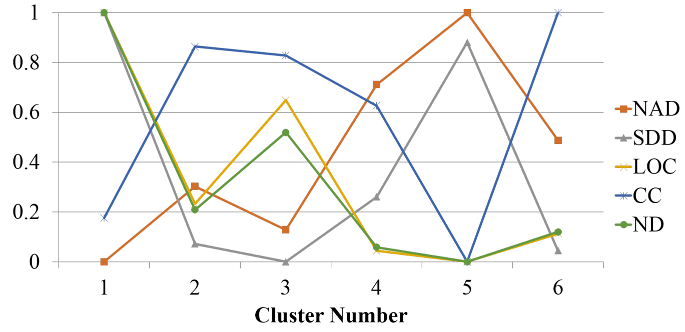
values of the average feature vector of each role in the network. As explained earlier, the feature vector includes the metrics normalized average degree (NAD), coefficient variation of the degrees of immediate neighbors (SDD), the clustering coefficient (CC), the locality index (LOC), and the normalized node degree (ND).

Similar to the first method, we have four roles for the GDP global trade network of countries.

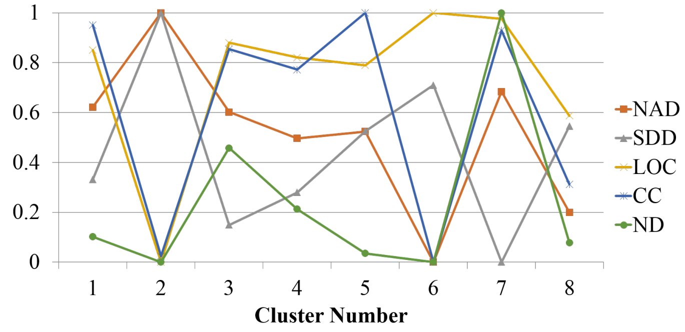
Dataset	Event	Time interval	Transition rules	Support	Z score
GDP	Shrink (4)	[23,40]	$4 \rightarrow 3$	100	14.70
	Shrink (1)	[9,29]	$1 \rightarrow 4$	25	12.1
	Emerge (3)	[12,22]	$4 \rightarrow 3$	126	8.3
	Emerge (2)	[26,32]	$3 \rightarrow 2$	80	6.3
USA airports	Growth (6)	[1,79]	$2 \rightarrow 6$	437	7.27
	Shrink (2)	[57,132]	$2 \rightarrow 3$	551	4.79
	Growth (3)	[142,175]	$1 \rightarrow 3$	1062	3.75
DBLP	Growth (4)	[1,11]	$0 \rightarrow 4$	6710	1.2
	Shrink (3)	[1,7]	$3 \rightarrow 0$	726	1.3
	Growth (7)	[6,11]	$0 \rightarrow 7$	196	0.2

Table 3.5: Description of some extracted events in the networks. Numbers in the parenthesis denotes the role number, holding the events

Table 3.5 shows some of the extracted events. In addition to the frequency count of every event in each dataset, we compared the result with randomized sequences in order to assert their significance. We built these random sequences by shuffling the order of role memberships of each node in the network. This way, the random dataset has the same number of nodes and roles. Since we have a dataset of role sequences for each network, we built 10 datasets of random sequences with the size as the number of nodes. We calculated average



(a) USA airport network.



(b) DBLP network.

Figure 3.3: The feature vector of the roles in selected networks.

and standard deviation of frequencies for each rule of events in the 10 datasets. Finally, we calculated the Z-score for the significance of each rule as compared to the randomized form. We show in the table the rules with the highest z-score.

Figure 3.4 graphically illustrates some of the events happening in each network in terms of transition of roles. Each graph includes all transitions of roles in the specified time interval. The label of nodes is the role number and the color of nodes shows the type of event happened to the role.

For example, the first event (a) in the GDP network describes the emergence of role three in time interval [12,22]. This graph says that nodes that have role three either come from role four or are the new nodes (role 0), just joining the network. By observing Table 3.5, we can see that the main reason for this event is the transition from role four. The same type of interpretation could be applied for the two other networks.

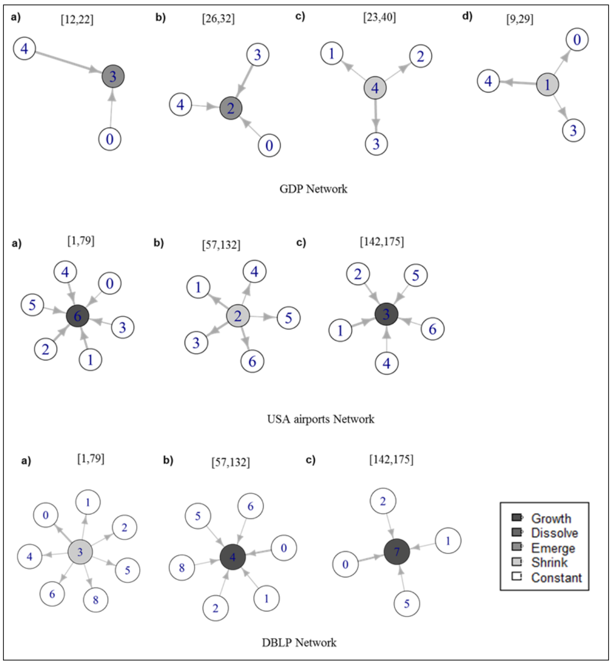


Figure 3.4: Some of the events happening in the networks.

3.4 Conclusions

Many networks are intrinsically dynamic and evolve over time. Discovering topological features in these networks is far from an easy task. In this chapter, we proposed two network characterization methods that considers both a static and a dynamic point of view in the framework of structural role mining. These methods are two-phase methodologies that automatically assign nodes to structural roles based on their local properties and extract events happening during the evolution of network. The static view provides a general description of the network through role assignment. Each role in the network is well characterized by the corresponding feature vector profiling. From a dynamic point of view, our methodologies discover evolution patterns either by defining time granularity or by defining event categories of roles, namely emerge, growth, constant, shrink and dissolve. The extracted events are described by some rules that depict the reason of each event and the flow of transition between roles.

We applied our methods to real networks to demonstrate and assess their capabilities. These methods explain the dynamics of networks by finding structural roles configuration and tracking its evolution by deriving events and their explanation in the form of transition rules. The rules show node transitions between roles and the the time interval of transition. The validity of the rules are evaluated in term of z-score of each event.

Our two proposed methods follow the same approach for role discovery in each time step of a dynamic network. However, each method studies the dynamics of structural roles from a different angle. The first method explained, in section 3.2, is more general and is capable of finding all evolution rules using different time granularities while the the second in section 3.3 is more exact and objective that only search for pre-defined events and detect the relative transitions.

Chapter 4

Pairwise structural role mining

Does the role of a node depend on its neighbors' roles? Do users at a similar structural position tend to connect to each other?

Pairwise dependencies (homophily), the tendency of users to connect with users of similar interests and social demography [MSLC01], is one of the sources of information for user behavior modeling [SR08, LAH07] and user classification [ZWY$^+$13]. However this is yet an open question for structural roles: *are structurally equivalent nodes more prone to have connections between themselves?*

In this chapter, we tackle this open question by studying the patterns of homophily for structural roles. We examine the structural role of users in a network regarding their ego properties and their connections to direct neighbors. We have a set of users (V) and a set of social connections between them (E). We want to infer a role configuration L over the social network of users, using two assumptions: 1) *ego-role dependence:* the structural role of users is correlated to their ego properties X and users in the same role k have similar feature vectors; 2) *pairwise role dependence*: the role of a user is not independent from its neighbors' roles. Before modeling the role mining problem using these two assumptions, we run a pilot experiment to study the pairwise relations between nodes across different structural roles. We show that pairwise relations can discriminate the role of users in the network. Our study suggests we should propagate role configurations only through certain connections rather than through all connections.

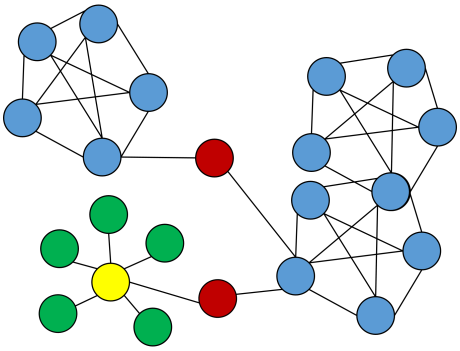


Figure 4.1: Pairwise dependency across structural roles, different colors correspond to different structural roles; the pairwise role dependency exists in some structural roles such as member-of-clique (blue nodes) but it does not hold on some others such as member-of-star (green nodes).

4.1 Pairwise dependency and structural roles

Pairwise dependency suggests that nodes with a similar structural position may have a tendency to have connections between themselves. Figure 4.1 exemplifies that, with the blue nodes (member-of-clique) having connections to other blue nodes. However, this it not the case for all types of structural roles. For instance, the green nodes (member-of-star) have no connections to other green nodes, as their structural features do not give origin to pairwise connections.

In this chapter our goal is to incorporate pairwise dependency of different structural roles in role mining framework. For that, we first examine how actually the pairwise relations are across structural roles by running a pilot experiment on a real social network.

4.1.1 Experiments configuration

For the pilot experiment, we use a very basic role mining method on a network to extract a set of structural roles and examine the pairwise dependencies between roles. We use the static role mining method explained in section 3.1 where the k-means algorithm is employed over the structural properties of nodes in the network, to group them into their respective roles.

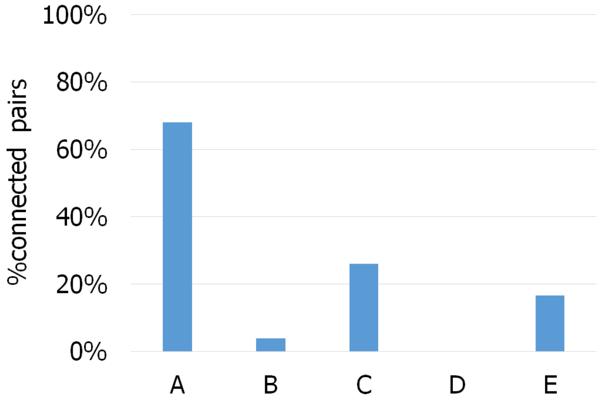


Figure 4.2: Pairwise dependency across structural roles in Digg social network; The percentage of connected users varies significantly across roles; A: “cliquey”, B: ”2nd periphery”, C: ”periphery-cliquey”, D: ”periphery”, E: ”local-star”. For example, 65% of users in role “A" are connected, but users of role “D" never connect.

4.1.2 Data

For this experiment we use a dataset from Digg *Digg*[1] social network. Digg is a news aggregator in which users can submit links to interesting news stories and they can rate these stories by voting on them. Users also can designate other users as friends. More specifically, each user has a list of followers (fans who follow him) and a list of followees (friends whom he follows). All activities are visible to her fans, including all stories he submitted or voted for. We use the Digg data collected by Lerman and Ghosh [LGS12] which contains the friendship network of users and all the posts submitted during one month, including the id, submitter id, voters for each post and the date of votes. This dataset includes 3,018,197 votes on 3,553 popular stories made by 139,409 users and the social network of active users (who have at least one vote) containing 71,367 users and 1,731,658 friendship links. We built our social network from active users and their connections, where active users are those who voted for at least one story.

4.1.3 Experiment results

We derived five structural roles in the network. Then we study the pairwise relations for different roles by counting the number of connected users in each role. Figure 4.2 depicts the results of this experiment for users in the social network of Digg. The percentage of connected users varies significantly across roles. For example, 65% of users in role “A” are

[1] http://www.isi.edu/ lerman/downloads/digg2009.html

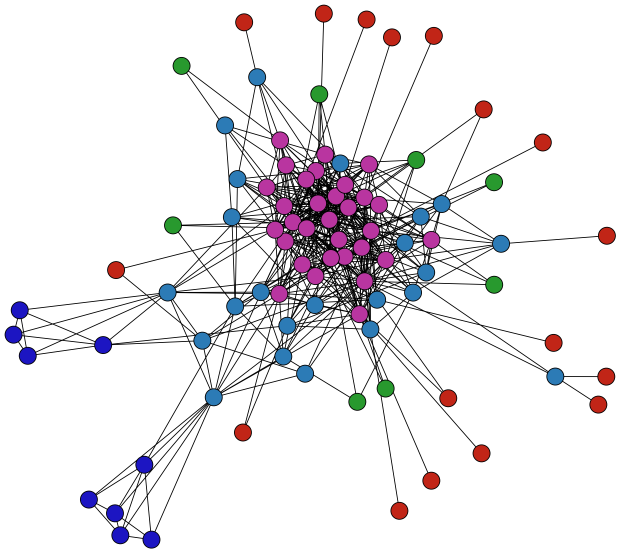


Figure 4.3: A subgraph of Digg social network, including active users in one information propagation process; color-coded regarding the structural roles; A: “cliquey” (purple nodes), B: ”2nd periphery” (blue nodes), C: ”periphery-cliquey” (dark blue), D: ”periphery” (red nodes), E: ”local-star” (green nodes) . The percentage of connected users varies significantly across roles. For example, 65% of users in role “A" are connected, but users of role “D" never connect.

connected, however users of role “D” are never connected. These two roles are depicted in Figure 4.3, with role “A” shown in orange and role “D” in yellow, corresponding to a subgraph of active users in an information cascade in a Digg social network including all of their connections to the rest of the network. We can clearly see that pairwise dependency is valid for role “A” but not for role “D”. Therefore, this dependency will be of great help in categorizing the users for some roles, but it can also be highly misleading for some other roles. In the next section we show how we can take advantage of pairwise dependency in role mining modeling.

4.2 Structural role modeling

In this section we explain our proposed method for role mining where both ego- and pairwise-role dependencies are considered following the framework of probabilistic graphical models [Pea88]. Our approach aims to detect groups of users that have the same structural properties and are socially connected. The likelihood of the data is higher when users in the same group have the same structural properties, and it is also higher when users have interactions.

We model these two dependencies in the framework of role mining. We first introduce the variables in the problem that are utilized for defining the objective function of our model. The first variable x_i represents the ego features of user u_i and it is derived by measuring a set of structural properties such as degree centrality and clustering coefficient. We define the set of ego features to be utilized in our model in section 4.2.5 . All features in this vector are normalized to the interval $[0, 1]$. The latent variable l_i shows the role label of user u_i and has value from 1 to K to indicate to which role the user belongs to. We quantify the pairwise dependency by variable λ_{kr} which measures the non-compatibility of roles k and r. Last, to represent the association between roles and ego features, we use an association variable μ_k for each role. Each dimension of this vector variable indicates the corresponding feature in the ego feature vector of x_i in the role k. Since we do not know which ego features are associated to role k, μ_k is an unknown vector and need to be learned.

For every two users u_i and u_j in the same role their ego feature vector x_i and x_j should be close on the dimensions designed by μ_k. Hence, by using a distance measure between ego feature vectors and association variable, we want to minimize:

$$\sum_{k=1}^{K} \sum_{u_i \in R_k} D(x_i, \mu_k) \tag{4.1}$$

where D is a distortion measure between users and R_k is the set of nodes with the label role k. Our model should also minimize the cost of pairwise role assignment to achieve a minimum role conflict between connected users:

$$\sum_{(u_i,u_j)\in E} \lambda_{kr}\mathbb{I}[(l_i = k, l_j = r)] \tag{4.2}$$

where λ_{kr} is the cost of non-compatibility of role k, r and $\mathbb{I}$ is the indicator function showing if the role labels of the connected users u_i, u_j are k and r. As discussed before pairwise role dependency is more important for some roles than for others. We tune $\Lambda = \{\lambda_{11}, ..., \lambda_{kk}\}$ in a way that it does not sacrifice the ego-role dependency for the sake of the pairwise dependency.

Our final objective function is derived from the linear combination of the two elements:

$$\begin{aligned} obj = \sum_{k=1}^{K} \sum_{u_i \in R_k} D(x_i, \mu_k) + \\ \sum_{(u_i,u_j)\in E} \lambda_{kr}\mathbb{I}[(l_i = k, l_j = r)] \end{aligned} \tag{4.3}$$

4.2.1 SR-Diffuse algorithm

In this section we introduce our algorithm to find the values of unknown variables such that they minimize Equation 4.3. We have three sets of unknown variables: 1) the role label l_i of user u_i, 2) the association vector μ_k for each role, and 3) the pairwise dependency cost λ_{kr} between the two roles k and r. Since the association vector as well as the role labels for the users are unknown, minimizing Equation 4.3 is an "incomplete-data problem", for which a popular solution method is Expectation Maximization (EM) [DLR77]. In the following we describe a soft role assignment (SR-Diffuse) algorithm which iteratively updates each set of variables.

The algorithm starts with an initialization of the three sets of variables. Afterwards, in the E-step, given the association vectors $\{\mu_1, ..., \mu_K\}$ and the pairwise dependency cost λ_{kr} for every pair of roles, every user is re-assigned to the roles that minimize her contribution to obj in Equation 4.3. In the M-step, the association vectors and the pairwise dependency cost are re-estimated from the role assignments $L = \{l_i, ..., l_N\}$ to minimize obj for the current assignment. Note that this corresponds to the generalized EM algorithm [DLR77], where the objective function is reduced but not necessarily minimized in the M-step.

Algorithm 1 SR-Diffuse

```
procedure SR-DIFFUSE($G = (V, E), K, \sigma$)
    $X \leftarrow egoFeatures(G)$
    $L^0 \leftarrow initialize(X)$

    $\Lambda \leftarrow updateVariables(L^0)$        $\triangleright \Lambda = \{\lambda_{11}, ..., \lambda_{KK}\}$
    while (not Converged) do
        $L^t \leftarrow roleAssignment(\{\mu_1, ..., \mu_K\}, \Lambda)$
        $\Lambda, \{f_1, ..., f_K\} \leftarrow updateVariables(L^t)$
        if $||L^t - L^{t-1}|| < \sigma$ then
            Converged $\leftarrow True$
        end if
    end while
    return $L^t$
end procedure
```

4.2.2 Initialization

To initialize the model, we applied the fuzzy k-means clustering algorithm [Yan05] to the dataset resulting in a partitioning of users into K clusters. We use this assignment to provide the values to the association vector, and to compute the variables relative to that assignment. These variables form the starting point for EM, which is then run to convergence.

4.2.3 Role assignment (E-step)

The assignments of users to roles are updated using the current estimates of the association vector and the pairwise dependency cost. In simple role assignment, when pairwise interactions of users is not considered, the E-step is a simple assignment of every user to the role representative that is nearest to it according to the distance function. In contrast, our model incorporates interaction between the users. As a result, computing the assignment of users to role representatives to minimize the objective function is computationally intractable in any non-trivial model [SWK03].

We follow the iterated conditional modes (ICM) [Bes86, ZBS00] approach, which is a greedy strategy to sequentially update the role assignment of each user, keeping the assignments of the other users fixed. This algorithm performs the role assignment in random order for all users. Each user u_i is assigned to the role label k that minimizes the user's contribution to the objective function. Optimal assignment for each user is the one that minimizes the distance between the users in the same role and maximizes the association between roles and ego features (first term of obj) with a minimal penalty for pairwise dependence assumption violations caused by this assignment (second term of obj). After all users are assigned, they are randomly re-ordered, and the assignment process is repeated. This process proceeds until no user changes its role assignment between two successive iterations. ICM is guaranteed to reduce obj or keep it unchanged (if obj is already at a local minimum) in the E-step [Bes86].

Overall, the assignment of points to roles incorporates pairwise supervision by discouraging assumption violations proportionally to their severity, which guides the algorithm towards a desirable role configuration over the network.

4.2.4 Update variables (M-step)

The M-step of the algorithm consists of two parts. First we discuss the update of the association vector μ_k for users in role k when labels $L = \{l_i, ..., l_N\}$ for all users are fixed. The association variables $\{\mu_1, ...\mu_k\}$ are re-estimated from users currently assigned to the roles to decrease the objective function obj in Equation 4.3. Each role association calculated in the M-step of the EM algorithm is equivalent to the expectation value over the points in that role, which is essentially their arithmetic mean.

$$\mu_k = \frac{\sum_{x_i \in L_k} x_i}{|L_k|} \tag{4.4}$$

The second set of variables that we discuss is the pairwise dependency cost λ_{kr} for the roles k and r. The main intuition for this variable is that users of certain roles tend to connect to each other but some others do not. Hence for fixed association vectors $\{\mu_1, ..., \mu_K\}$ and role assignment $L = \{l_i, ..., l_N\}$, we estimate the pairwise dependency cost λ_{kr} as follows:

$$\lambda_{kr} = \frac{|u_i : l_i = k|.|u_i : l_i = r|}{|(u_i, u_j) \in E : l_i = k, l_j = r|} * \alpha \tag{4.5}$$

where the denominator measures the number of pairs of (k, r) in the network and it is normalized by the number of connections if these roles where always connected. The basic idea is that cost of having same role for connected nodes is higher if it is a rare case in the network.

To complete the model parameterization, we need to specify the value of α, the variable used in Equation 4.5 to represent the strength of the preference towards assigning connected users to the same role. We experimented with a range of values for α for both data sets, measuring both the number of connections in each role and the coherence of the clusters with respect to the structural properties. We evaluated the structural coherence of a role as the average distance between every pair of users that were assigned to the role. As expected, increasing α results in a larger number of connections among users in the same role. SR-Diffuse results in roles configuration consistent with the pairwise role dependence assumption, while not sacrificing the structural properties quality. This parameter also helps in finding the appropriate number of roles for a network, as we discuss in more detail in section 4.2.6.

4.2.5 Ego features

In this section we define the ego feature vector for the users. It is possible to use a different feature set for role mining such as local features [CSRPar] or recursive feature aggregation [HGL$^+$11] We selected the described egonet features in section 2.2.1.1 of chapter 2 as structural properties of users which has been shown to be correlated to social roles of users [CRHK09, CSRPar] as follows:

- the normalized node degree (ND)
- the normalized average degree (NAD)
- the standard deviation of degree (SDD)
- the clustering coefficient (CC)
- the locality index (LOC)
- the common neighbors (CN)
- the eigenvector centrality ($eig - cntr$)

4.2.6 Determining number of structural roles

The number of roles is one of the challenges in role mining as discussed in chapter 2. Our role mining method solves this issue by initializing the number of roles to a relatively large number ($N/2$) and when it stops the non-empty roles are the final roles. The final number of non-empty roles is determined by the value of α in Equation 4.5. We study the effect of value of α by measuring the quality of roles in two terms: *isolation* and *compactness*. Isolation assesses how well are roles separated by calculating the distance between the centers of roles and compactness assess the coherence of roles by measuring the distance between users in the same role [BL97]. We calculate the quality score of discovered roles by:

$$\begin{aligned} qualityScore = Isolation - Compactness = \\ \underset{\forall r,k \in [1,K]^2}{min} dist(\mu_k, \mu_r) - \\ \underset{\forall k \in [1:K]}{mean} \underset{x_i \in L_k}{max} dist(\mu_k, x_i) \end{aligned} \tag{4.6}$$

The higher score shows higher quality for a role set as it shows roles are well separated by high value of isolation and have high coherency by low value for compactness component. We find the appropriate number of roles, K, by varying the value α as long as it improves the Equation 4.6.

4.3 Experiments

As discussed in section 2.4.3, evaluating of role mining methods is a challenging task. In this chapter, we demonstrate the efficacy of SR-Diffuse through user classification in information cascades. An information cascades is a process of spreading information, in which nodes cause connected nodes to be activated in terms of reposting a piece of information with some probability [LGS12]. For S cascades we label involved users in each cascade regarding the class label definition in section 4.3.2. The classification task is to predict the labels of users in a cascade based on the role membership matrix. We use logistic regression for this purpose. We measure the predictability of the discovered roles by SR-Diffuse and compare it to three baseline methods, each evaluating a different set of properties in the network.

4.3.1 Data

Throughout this entire section we will be using two different datasets, coming from two well known and established internet communities: *Digg*[2] and *Flickr*[3]. Both include a static social network with social relationships between users and a dynamic evolving network describing information propagation.

[2] http://www.isi.edu/ lerman/downloads/digg2009.html

[3] http://socialnetworks.mpi-sws.org/datasets.html

Table 4.1: Summary of datasets.

Data	#Users	#Objects (story/photo)	#Links	Network time interval	Time granularity
Digg	71,367	3,553	1,731,658	5 years	three months
Flickr	914,400	4,000	18,595,048	2 years	one month

Flickr is a popular photo and video hosting website with a large community of users. We use data collected by Cha et al. [CMG09a], which includes a social friendship network of users and information propagation from one user to another. The associated mechanism is similar to Digg, but instead of URLs, photos are shared and voted. This dataset contains data of 104 days (starting Nov 2, 2006) on 34,734,22 favorite markings of 11,267,320 photos. The social network has 1,620,392 users and 33,140,018 edges. We randomly sampled 4000 photos from those which number of favorite marking is higher than 100. The social network includes all users who have marked the selected photos as favorites and all their connections in the original data.

The second dataset we used is Digg social network as described in section 4.1.2. Table 4.1 summarizes the statistics of the both datasets.

4.3.2 User classification in information cascade

Role mining in the networks gives an abstraction of the network in terms of matrix membership of users which can be used for several applications including node classification. In this section, we first define a set of social classes for users in an information cascade and then study how the defined classes can be predicted.

In an information cascade, the capability of users in spreading information is of great interest. An important parameter for categorizing users in a cascade is their effect on the network which we measure as the consequence of user's action. The time interval of involvement of users in the process is also important as the late adopters are not of interest for diffusion modeling and spreading the story. Here, we define a new classification for users in an information cascade by two factors.

1. time of action: we divide the lifetime of a cascade in to three phases:
    - slow growth: the time slot when the cascade size is less than 5% of final size
    - explosive phase: when cascade size grow from 5 to 90% of final size.

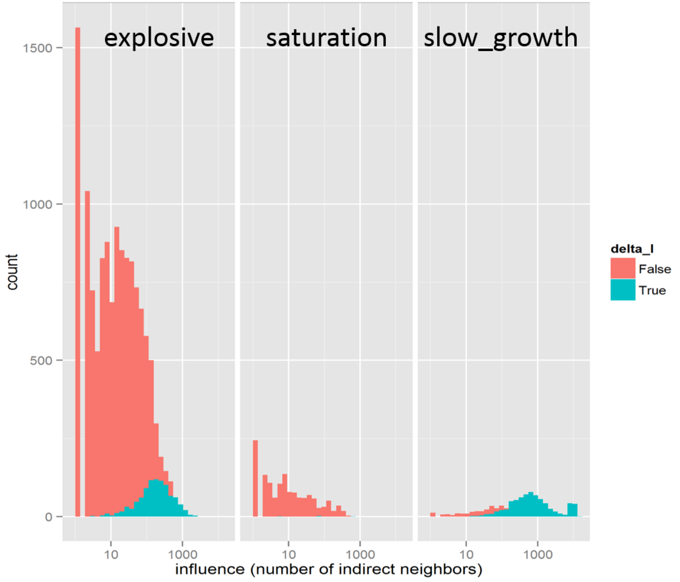


Figure 4.4: Influence distribution of users over time.The blue distribution belongs to the users with $l(t + \tau)/l > 1$, who could influence the network beyond their immediate neighbors. In the “slow growth” phase these users have larger degree (number of immediate neighbors) .

- saturation phase: when cascade size is above 90% of its final size

2. consequence of action: We use the the multiplicative number of node i, which is the quotient of the number of listeners reached one time step after i showed activity, $l(t + \tau)$, and the number of nearest listeners of u_i, i.e., those who instantaneously received its message, $l(t)$ (which is given by the number of followers of i that are involved in the cascade). Thus, the ratio $l(t+\tau)/l$ measures the multiplicative capacity of a node: $\delta_l = l(t + \tau)/l > 1$ indicates that a user has been able to increase the number of listeners who received the message beyond its immediate followers.

Figure 4.4 shows the distribution of influence of users at defined time phases. The blue distribution shows influential users, and the red one belongs to those that were not able to affect network beyond their 1-hop neighborhood. As we can see, not all the early adopters in the “slow growth” phase are influential enough to affect users for further voting. The red distribution has higher frequency but lower influence mean comparing to the blue one. Regarding the aforementioned factors and Figure 4.4, we categorize users that are active in a cascade into six groups or classes. Active users are those that vote for a story or repost a story:

- initiators: active users in slow growth phase with $\delta_l > 1$.

- promoters: active users in explosive phase with $\delta_l > 1$.
- early adopters: active users in slow growth phase with $\delta_l < 1$.
- common users: active users in explosive phase with $\delta_l < 1$.
- late adopters: active users in saturation phase with $\delta_l > 1$.
- passives: active users in saturation phase with $\delta_l < 1$.

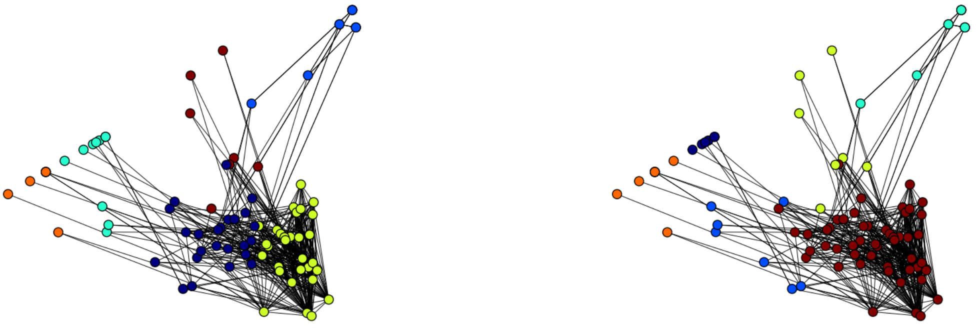
These six groups constitute our class labels and we call them social roles to differentiate them from structural roles that we have from the role mining framework. In this chapter we investigate how social roles correlate to the structural roles and we demonstrate the predictability of our role mining method through predicting social roles in a cascade.

4.3.3 Experiment configuration

We first determine the suitable number of roles in a network regarding the method explained in section 4.2.6. Figure 4.6 shows the quality of discovered roles for different values of α. As we can see, the worst quality belong to the setting with $\alpha = 0$ which is basically when pairwise dependency has zero effect in the role mining. This demonstrates that SR-Diffuse improves role mining results by incorporating the pairwise dependency. SR-Diffuse specially improves the quality of role sets in the network when the roles are very similar and relying only on structural features is not enough for learning the roles.

Figure 4.5 shows a subgraph of Digg social network, where nodes are positioned regarding their first and second principle component of the matrix of nodes ego features. In this figure nodes with similar ego features are located closely. In the network (a) nodes are color coded regarding their role from k-means algorithm while nodes in the network (b) are color coded by their roles discovered from SR-Diffuse. As we can see for the same number of roles, different role configuration is derived by two methods. SR-Diffuse puts connected nodes that are close regarding ego vectors in the same role while k-means can not; green and dark blue nodes in network (a) are placed in the same group (dark red) by SR-Diffuse and cyan nodes in network (a) are divided into two roles (dark and light blue) in network (b).

From Figure 4.6, we can see that SR-Diffuse finds the best roles configuration on Digg social network when $\alpha = 56$ and on Flickr network $\alpha = 72$. With this configuration the number of roles that SR-Diffuse found on these networks are respectively 8 and 11. We use the same number of roles for the baseline methods. Next we explain how discovered roles can predict social roles of users in an information cascade.



(a) Color coded by k-means

(b) Color coded by SR-Diffuse

Figure 4.5: A subgraph of Digg social network, including active users in one cascade; color-code regarding the structural roles.

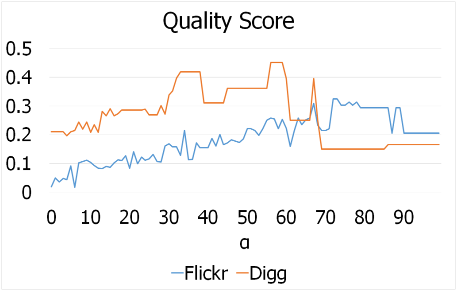


Figure 4.6: The quality of discovered role set by SR-Diffuse for different values of α.

We select S disjoint cascades that do not have any active users in common. We measure the ego properties of the N active users in the cascades and then learn structural roles of users by a role mining method (SR-Diffuse, pair-means and c-means). This gives us the role membership matrix of users which we use as predictor to build the classifier using logistic regression. In order to be able to evaluate the predictability and generality of discovered roles we use 50% of users to build the role membership matrix and put the rest aside as the test set. We use the role membership matrix of users in the train set to build the classifier and the evaluation result is derived from the classification of users in the test set.

Table 4.2 demonstrates the evaluation results of SR-Diffuse and the baseline methods. We measure the performance of each method in terms of F-score for the predicted roles in the test set. F-score is the harmonic mean of precision and recall which are respectively equal to $\frac{|p \cap r|}{|p|}$ and $\frac{|p \cap r|}{|r|}$ for the predicted role p with reference to actual role r. We can see that SR-Diffuse can better predict roles of users in an information cascade

We compare the predictability of discovered roles by SR-Diffuse to three baseline methods:

the first one evaluates the effect of pairwise role dependence assumption, the second one evaluates the effect of ego properties of users on the roles and the third one compare the predictability of structural roles to ego properties:

- pair-means: this method uses pairwise role dependence for cluster assignments, but does not perform distance learning; it applies majority votes on the labels of neighbors of a user to infer its role. This method is initialized by clustering a subset of users using the fuzzy k-means algorithm and then the role labels for the rest of the users are assigned by majority votes.
- c-means: the fuzzy k-means algorithm over structural properties is used for role discovery.
- ego-feat: this method uses only the ego features of users, as described in section 4.2.5, to make the prediction model.

Table 4.2: Performance of SR-Diffuse in classifying users in information cascade in comparison to baseline methods.

Digg	F1	precision	recall
SR-Diffuse	0.50	0.67	0.41
c-means	0.44	0.52	0.39
pair-means	0.46	0.67	0.36
ego-feat	0.29	0.76	0.18

Flickr	F1	precision	recall
SR-Diffuse	0.46	0.58	0.39
c-means	0.44	0.61	0.35
pair-means	0.40	0.57	0.31
ego-feat	0.33	0.62	0.23

Table 4.2 reports the classification performance of discovered roles by SR-Diffuse comparing to the baseline methods in terms of F1, precision and recall. We can see that the worst performance (lowest F1) belongs to the ego-feat method. However, its precision is the highest one. This shows that the ego features are good indicators for social roles of users in in information cascade. The recall is low, which suggests that ego features are not enough for predicting roles. The classifier performs better when the structural role membership is used as the predictor instead of ego features. As we can see from the table, we have better

classification performance for all three role mining methods (SR-Diffuse, c-means and pair-means) over the ego-feat method. Overall, the role configuration discovered by SR-Diffuse is a better classifier for social roles in information cascade as we have the best classification results from the classifier trained over this role membership matrix. This suggests that the combination of ego features and pairwise dependencies can improve the quality of role mining results and better detect existing structural roles in the network.

4.4 Conclusions

In this chapter, we proposed a new method for structural role mining by incorporating pairwise dependencies along side with ego features of users. We showed how structural compatibility varies across different structural roles and devise a method to take advantage of this property for discovering some of structural roles and avoiding the deception for the others. Our method is capable of finding the roles membership of users regarding their structural features and pairwise dependencies. It iteratively assigns users into structural roles in such a way that the derived roles set has the highest possible coherency in terms of including the most similar users and has the least non-compatibility of roles in the neighborhood of each user. This algorithm automatically finds the appropriate number of roles in a network by controlling the pairwise dependency parameter.

In this chapter, we also explored how influential users modeling in information cascade can benefit from structural role mining in a network. We defined a set of class labels for active users in information propagation events on a social network based on their influence and time of action and then used structural roles membership of users to predict their class labels in an information cascade. We have shown that the structural roles obtained by our method are better predictors for social classes of users, when compared to a set of baseline methods.

Chapter 5

Evolutionary structural role mining

How is the temporal behavior of nodes reflected in their structural roles? How can we detect dynamic roles of nodes?

In this chapter, we propose a new method to determine structural roles in a dynamic network based on the current position of nodes and their historic behavior. We develop a temporal ensemble clustering technique to dynamically find groups of nodes, holding similar tempo-structural roles. We compare two weighting functions, based on age and distribution of data, so that we incorporate the temporal behavior of nodes in the role discovery. We define *evolutionary role extraction* as the problem where a sequence of graph snapshots are given and the goal is to find roles of active nodes at the current time. These roles must reflect the structure of the network at the current time and must be consistent with existing roles in the network, extracted at previous times.

5.1 Evolutionary role mining

Most of the complex networks are dynamic where a group of nodes join the network or some relations are altered. All these changes in the structure of networks alter the position of nodes within their neighborhood and provoke changes their structural roles. In this chapter, we describe our proposed method for evolutionary role mining. The discovered roles by this method represent the temporal behavior of nodes. For example, in a co-authorship network, we may want to find the evolutionary role of pioneers in a topic and track their behavior. This helps to derive a profile of the existing dynamic roles in a dynamic network. We define the evolutionary role mining as a problem where a sequence of graph snapshots are given and, the objective is to find structural roles of nodes such that : 1) the role of nodes at current time should be close to previous time, if the connectivity of nodes does not deviate from previous time points; 2) the set of roles must be modified to reflect the new structure, if the structure of the network changes significantly. Our framework is an online algorithm where a set of roles for the network at time t is obtained before having access to the networks at next time steps.

A possible solution to this problem is employing evolutionary clustering [CKT06]. This method is an incremental process where clustering C_t is built up on C_{t-1}, and the cost function of the clustering algorithm is evaluated based on the original similarity feature space. Both of these characteristics of evolutionary clustering make it computationally expensive.

In this chapter, for the first time we use ensemble clustering [SG03] for temporal data to extract the grouping of data regarding their feature set and their history. Ensemble clustering combines multiple partitionings of a set of objects without accessing the original features. It has been shown that ensemble clustering can improve the results by aggregating the different partitionings of objects [TLJF04, FJ02, GMT05, SG03]. Streh and Ghosh indicated the improvement of the clustering results robustness and the possibility of using distributed computing as two of the main motivations for using this method this method [SG03].

We first introduce the notations that we will be using throughout this chapter. We have a dynamic social network $G_t = (V_t, E_t, X_t)$ where $V_t = \cup_{i=1}^{t} V_i$ is the set of unlabeled users at time t, E_t represents the set of connections in the network and D_t is the set of structural properties of users. Suppose the set of labels $C_t = \{R_1, ..., R_K\}$ represents the K groups of users at time t.

The goal is to find the labels of users from their structural properties over time. We propose a dynamic ensemble clustering [TLJF04, FJ02, GMT05, SG03] framework such that the partitioning of users represents the current role set in the network at time t and is also consistent with the historical information of users in previous time steps. In our method, the clustering of L_t is derived from the aggregation of $C = \{C_1, ..., C_t\}$, a set of clusterings over time. We define a new similarity metric between users based on how similar they have been clustered over time. The partitioning of users based on this new metric gives the actual roles of users at the current time.

The pseudo-code of our evolutionary role mining method (ERM) is given in Algorithm 2. It takes as input: 1) G_t, the dynamic graph where edges are time stamped; 2) K, number of roles to extract; 3) $weightingfun(C)$, a weighting function to incorporate temporal behavior in partitioning; 4) $clusteralgo(simmatrix, K)$, an algorithm to partition users into K roles based on the calculated $simmatrix$ (a similarity matrix as detailed in section 5.1.2).

Algorithm 2 Evolutionary Role Mining (ERM)

```
procedure ERM($G_T = (V, E_T), K, weightingfun(C), clusteralgo(simmatrix, K)$)
    for $t$ in $1 : T$ do
        $X_t \leftarrow localProperties(G_t)$
        $C_t \leftarrow k - means(X_t, K)$
        $C \leftarrow C \cup C_t$
    end for
    $A \leftarrow weightingfun(C)$
    for $t$ in $1 : T$ do
        $simmatrix \leftarrow simmatrix + pairwiseSimilarity(V, C_t) * \alpha_t$        $\triangleright \alpha_t \in A$
    end for
    $L_T \leftarrow clusteralgo(simmatrix, K)$
    return $L_T$
end procedure
```

5.1.1 Local properties measurement

The first step in evolutionary role mining is to build clusters from structural properties of users X_t for all $t \in [1 : T]$. This clustering process is derived by applying the k-means clustering algorithm on X_t, where the euclidean distance between observations and centroids is minimized. We selected the same structural properties as explained in 3.1 plus two more features (common neighbors and eigenvector centrality) as follows:

- Normalized node degree (ND)
- Normalized average degree (NAD)
- Standard deviation of degree (SDD)
- Clustering coefficient (CC)
- Locality index (LOC)
- Common neighbors (CN)
- Eigenvector centrality ($eig - cntr$)

This step gives us a set of data clusters, $\{C_1, ...C_T\}$, which are then used for finding the role of users at time T using evolutionary ensemble clustering.

5.1.2 Temporal nodes similarity

We define a new similarity matrix of users based on their co-clustering occurrence at previous time steps. The similarity matrix is an $N \times N$ matrix for N active nodes at the current time step. For two users u, v, if $C_i(u) = C_i(v)$ then $\mathbb{I}_i(u, v) = 1$ and the total similarity of u, v is:

$$similarity(u, v) = \sum_{i=1}^{t} \mathbb{I}_i(u, v) * \alpha_i \tag{5.1}$$

where $\mathbb{I}_i$ is an indicator function which tells if two users are in the same group in clustering i or not, and α_i is the weight of the clustering at time i. The value of α_i is determined by a weighting function as will be discussed in next section.

5.1.3 Weighting functions

The network dynamics are embedded in ERM by incorporating a weighting function that assigns more importance to some temporal data, and gives less weight to other data. We need a mechanism to identify those clusterings that are not consistent with the current clustering due to noise or concept drift. The common approach for these cases is to use either *temporal weighting*, or a *sliding window*. Another method for weighting the clustering is using the data distribution instead of the arrival time of data. We use two different scenarios to model the temporal behavior of data in clustering ensemble. The sliding window method is excluded from our study since it requires multiple cluster aggregation for deriving grouping of data at each time point. In addition, this method generates several clusterings at each window which need to be corresponded. All these make it less applicable comparing to the two other methods.

Temporal weighting

Based on Cormode et. al. [CSSX09], we define a new exponential time decaying function, called temporal weighting (TW), to be used in ERM:

$$\alpha_i = (1 - \theta)^{T-i}, \ for\ i = 1\ to\ T. \tag{5.2}$$

where θ is a decaying factor to emphasize on recent data.

This function defines the probability that historic data is still valid for learning the roles at the current time. The underlining idea of this weighting function is that older data is less relevant than current data, so a lower weight is assigned to older data. Different functions can be defined in this group but the general properties that all must hold are:

1. $0 \leqslant \alpha_i \leqslant 1$ for all $i \in [1, T]$;

2. $\alpha_i < \alpha_j, i < j \leq T$

3. $\alpha_t = 1$.

Data distribution

We define a data distribution weighting (DDW) function that assigns weights to data based on its similarity to the current data. We use the distance of two clusterings to define the weights. The distance of current clustering C_t and C_i is defined as the number of objects they have clustered differently [GMT05]. The distance between two nodes u and v for two clusterings t and i is:

$$d_{u,v}(C_t, C_i) = \begin{cases} 1, \text{ if } C_t(u) = C_t(v) \text{ and } C_i(u) \neq C_i(v), \\ \text{ or } C_t(u) \neq C_t(v) \text{ and } C_i(u) = C_i(v) \\ 0, \text{ otherwise} \end{cases} \tag{5.3}$$

Then the distance of clusterings is measured as:

$$dist(C_t, C_i) = \sum_{u,v \in V_t} d_{u,v}(C_t, C_i) \tag{5.4}$$

As a consequence of the previous Equation 5.4, the weight of clustering at time i is:

$$\alpha_i = 1 - Norm(dist(C_t, C_i)) \tag{5.5}$$

where $Norm(dist(C_t, C_i))$ is the value of distances normalized to the interval $[0, 1]$.

The main idea behind this type of weighting function is different from the temporal weighting function. Here, the validity of data is not assessed by its age but by its actual similarity to the current data. In this method, the older clustering that groups objects more similar to current data is more important than the recent clustering that does not. In other words, if $C_{t-k}(u) = C_t(u)$ and $C_{t-j}(u) \neq C_t(u)$ where $k > j$ then $\alpha_{t-k} > \alpha_{t-j}$ where C_t is the clustering at time t and α_t is the weight of clustering at time t. This method has also been used for weighting models to build ensemble classifiers [WFYH03].

5.1.4 Cluster ensembles

We obtain a set of clusterings of users for all time steps and combine these individual clusterings to obtain an ensemble clustering that categorizes users into social groups. We used three clustering algorithms on the derived similarity matrix derived from Equation (5.1) to find the ensemble clustering.

We modified the *hypergraph partitioning algorithm (HGPA)* by Strehl and Ghosh [SG03] to use the weighted similarity metrics. This method re-clusters the objects using the hypergraph built upon the clusterings. In this method the hypergraph partitioning package HMETIS [KAKS97] is used to partition the hyper graph.

Spectral clustering is tipically used for graph partitioning problems where a graph-based measure, such as the normalized cut, is to be minimized. This algorithm clusters objects based on the eigenvectors of their similarity matrix. For the nodes and their similarity, measured by Equation (5.1), the graph Laplacian ι is built: $\iota = S - W$ where S is the degree diagonal matrix of the similarity graph of nodes and W is the similarity matrix of data. Then the first k eigenvectors of ι are calculated. Finally the clustering is derived by applying k-means on a matrix, built from concatenation of the first k eigenvectors as columns [VL07].

Another possibility for aggregating the clusterings over time is *agglomerative hierarchical (Agglo)* [Joh67]. This algorithm initially puts all objects in individual clusters then iteratively merges pairs of clusters either until deriving the defined number of clusters or until merging all the objects into one single cluster.

5.2 Experimental Results

We applied ERM on real world data sets to evaluate its performance. We used three coauthorship networks (DBLP, Genetics and Biochemistry [WSP07]) and the network of Internet routing system [LKF05a] to find evolutionary roles and demonstrate the performance of the proposed evolutionary clustering.

5.2.1 Data

- The DBLP dataset contains the publications of the proceedings of 28 conferences related to Data Mining, Databases and Machine Learning from 1997 to 2006.

- The Genetics dataset contains articles published from 1996 to 2005 in 14 journals related to genetics and molecular biology.

- The Biochemistry dataset contains articles published from 1996 to 2005 in 5 journals related to biochemistry.

- The autonomous systems network (AS) is comprised of the internet routing system. This is a daily dataset from SNAP network data collection[1]. We aggregated the daily instances to derive monthly graphs from November 1997 to August 1998.

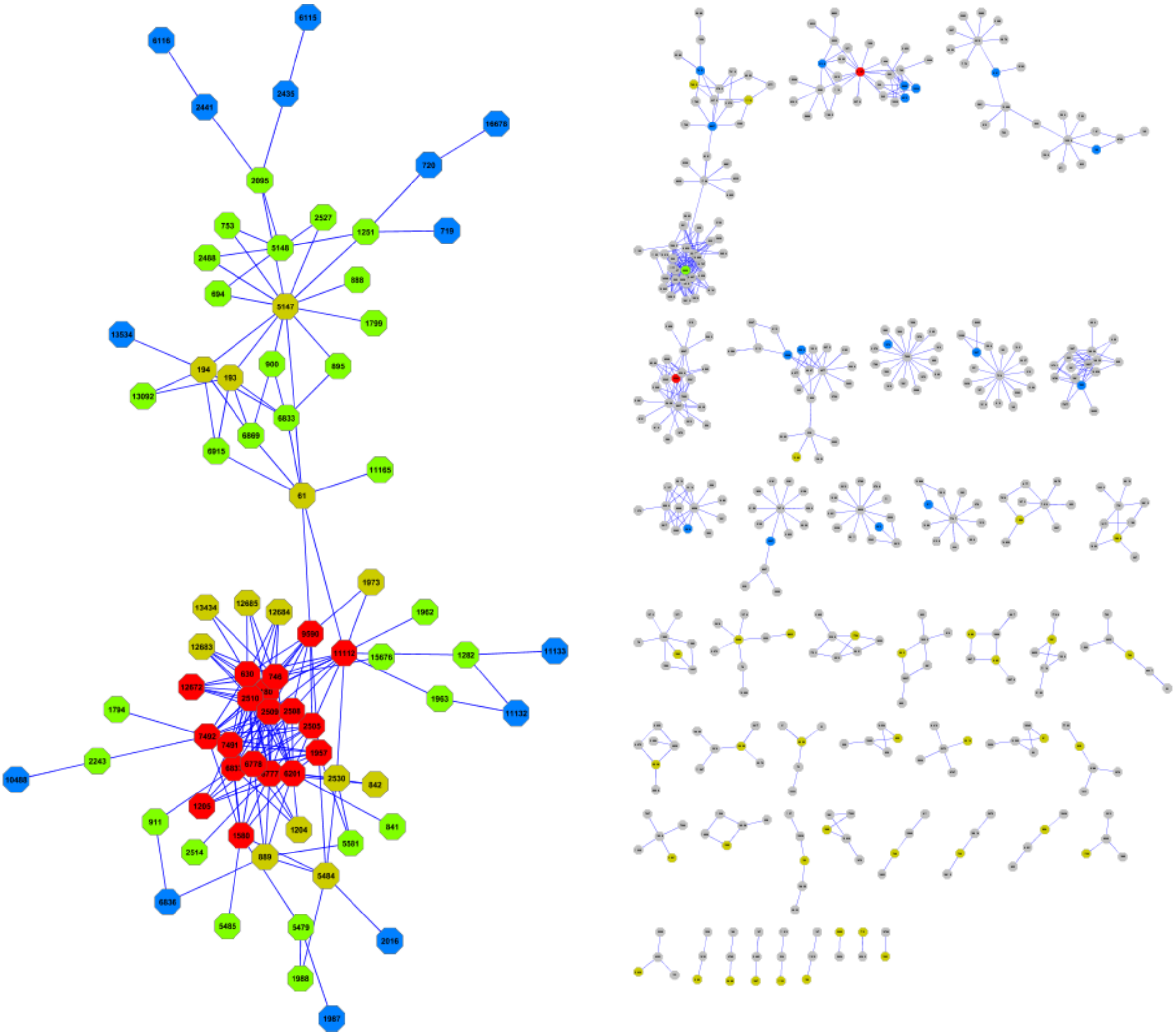
5.2.2 Results

We defined two baselines to compare the results against. The first baseline (CL) stacks all data up to current time t to find the clustering of data. The clusters are derived by applying the k-means algorithm on the stacked matrix. This is the general approach in evolutionary clustering where all data is available. In addition, previous studies of dynamic role discovery employ this approach [CSR11, RGNH12]. The second baseline (CLs) clusters data at each time step independently using k-means and discard historic data to derive the roles in the current snapshot of the network.

Figures 5.1 and 5.2 illustrates the second largest connected component of the DBLP network in 2002 and the connectivity structure of the same nodes in 2003. Nodes are colored by their roles, identified by our proposed method and the CL baseline method. As we can see from the figures, roles of nodes identified by our evolutionary method more accurately represent the actual position of nodes in the network in 2003. For example, all less connected nodes in very sparse neighborhoods are colored the same (dark yellow) in Figure 5.1a while we can see in Figure 5.2a the same nodes have various labels, determined by baseline method.

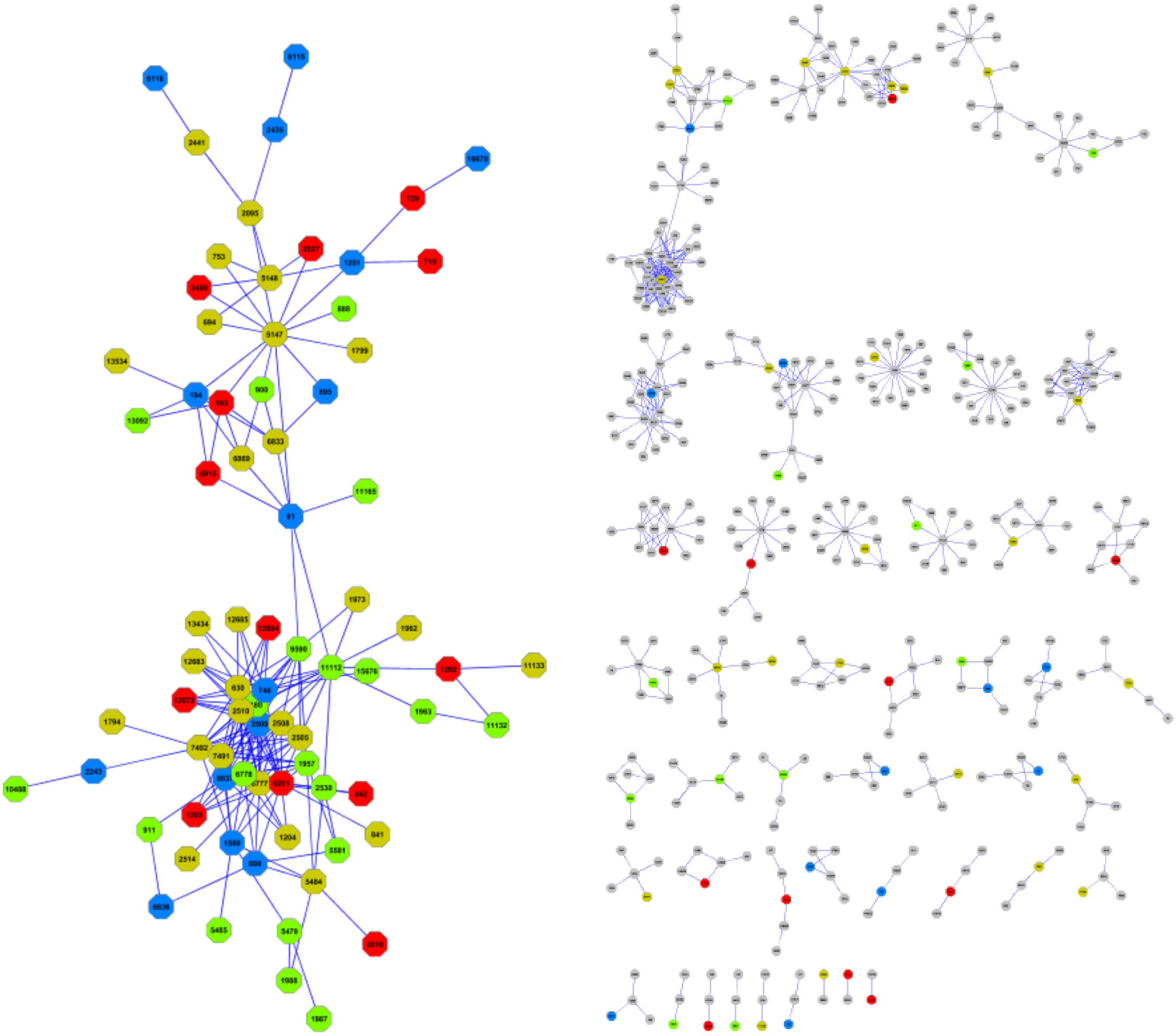
To compare the performance of the algorithms, we measure the snapshot cost which is the quality of clustering on the current data. We use the modularity metric proposed by Newman [NG04] to assess the quality of clustering. This metric evaluates the community structure in a network where a $K \times K$ matrix is built for K clusters and every element d_{ij} represents the fraction of edges that link nodes between clusters i and j and d_{ii} is the fraction of edges within cluster i. We use similarity metrics of nodes D_t to build a similarity graph where edge e_{ij} is weighted by the similarity d_{ij} between node i and j. We modify the modularity measure for weighted network of nodes' similarity by having d_{ij} representing the sum of the edges weights between two clusters, instead of the sum of the number of

[1] http://snap.stanford.edu/data/index.html



(a) Color-code by role of nodes, identified by proposed method

Figure 5.1: The second largest connected component of DBLP network in 2002 (left panel) and neighborhood of the same nodes in 2003 (right panel). The colors depict roles of node in the network, identified by ERM. In 2002 the identified roles are almost the same as the result of baseline method but in the consecutive time step, ERM method can detect the roles of nodes more accurately and coherently.



(a) Colors are determined by the CL baseline method

Figure 5.2: The second largest connected component of DBLP network in 2002 (left panel) and neighborhood of the same nodes in 2003 (right panel). The colors depict roles of node in the network, identified by the baseline method. In 2002 the identified roles are almost similar to the result of ERM method but in the consecutive time step, ERM method can detect the roles of nodes more accurately and coherently.

edges used in the original definition, and d_{ii} is the fraction of the sum of the edge weights within a cluster by the total edge weights. The modularity is calculated as follows:

$$modularity = \sum_{i=1}^{k}(d_{ii} - \sum_{j \in 1:k, j \neq i} d_{ij}) \tag{5.6}$$

The main aspect of evolutionary role extraction is to increase consistency of clustering with previous time steps. We use historical cost to measure the smoothness in the transitions between time steps. The historical cost quantifies the degree to which the proposed algorithm can enhance temporal smoothness. We assess the consistency of successive clusterings by using normalized mutual information (NMI) [SG03].

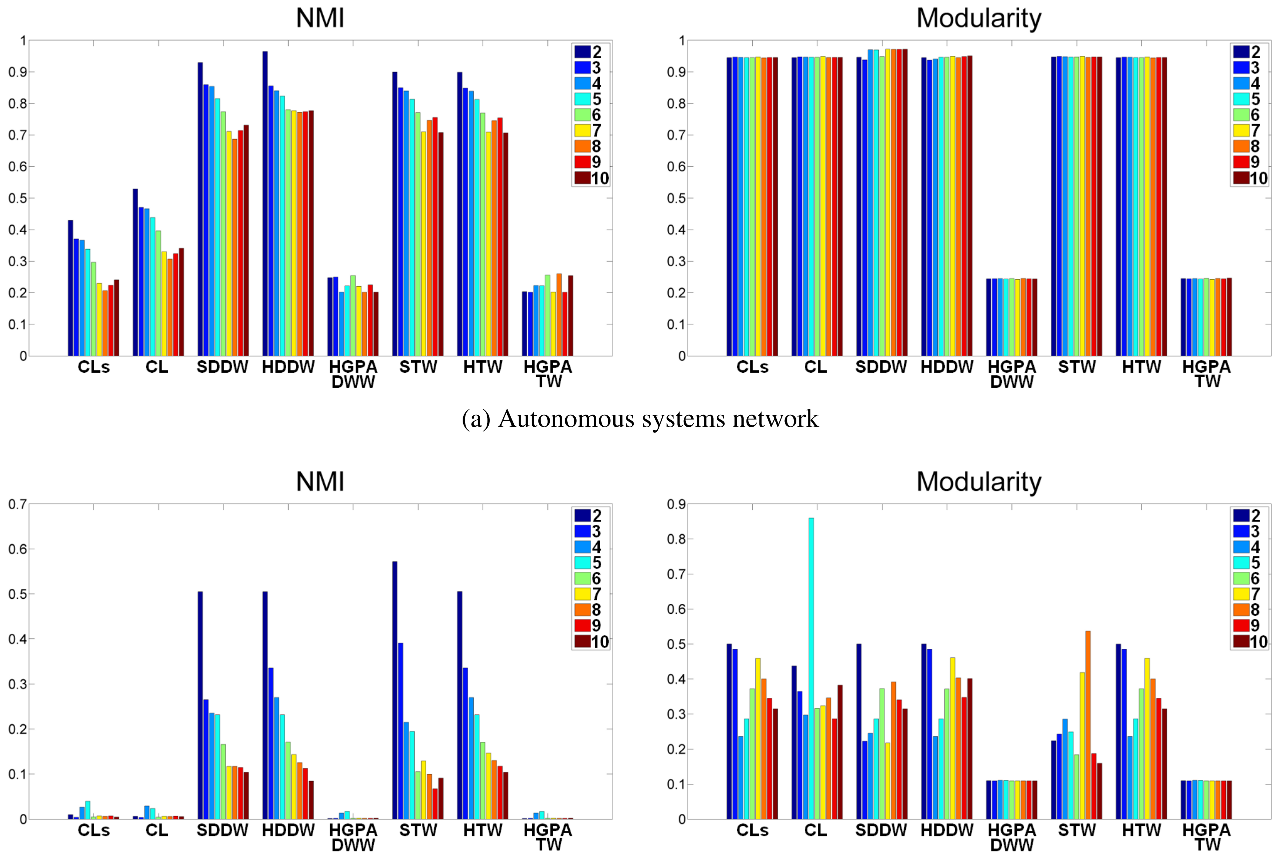
Assume r groupings denoted as $R = \{\lambda^q, q \in \{1, ..., r\}\}$, then the normalized mutual information between two groupings λ^a and λ^b is estimated as:

$$\phi^{NMI}(\lambda^a, \lambda^b) = \frac{\sum_{h=1}^{k^a}\sum_{l=1}^{k^b} n_{h,l} log(\frac{n.n_{h,l}}{n_h^a n_l^b})}{\sqrt{\sum_{h=1}^{k^a} n_h^{k^a} log(\frac{n_h}{n}) \sum_{l=1}^{k^b} n_l^{k^b} log(\frac{n_l}{n})}} \tag{5.7}$$

where n_h^a is the number of objects in cluster C_h according to λ^a, and n_l^b the number of objects in cluster C_l according to λ^b, $n_{h,l}$ denote the number of objects that are in cluster h according to λ^a as well as in group l according to λ^b.

In Figure 5.4 the performance of different weighting functions and algorithms on each data set is compared. Each panel demonstrates the NMI and Modularity of the results on used data. For both evaluation metrics the higher values indicate a better performance.

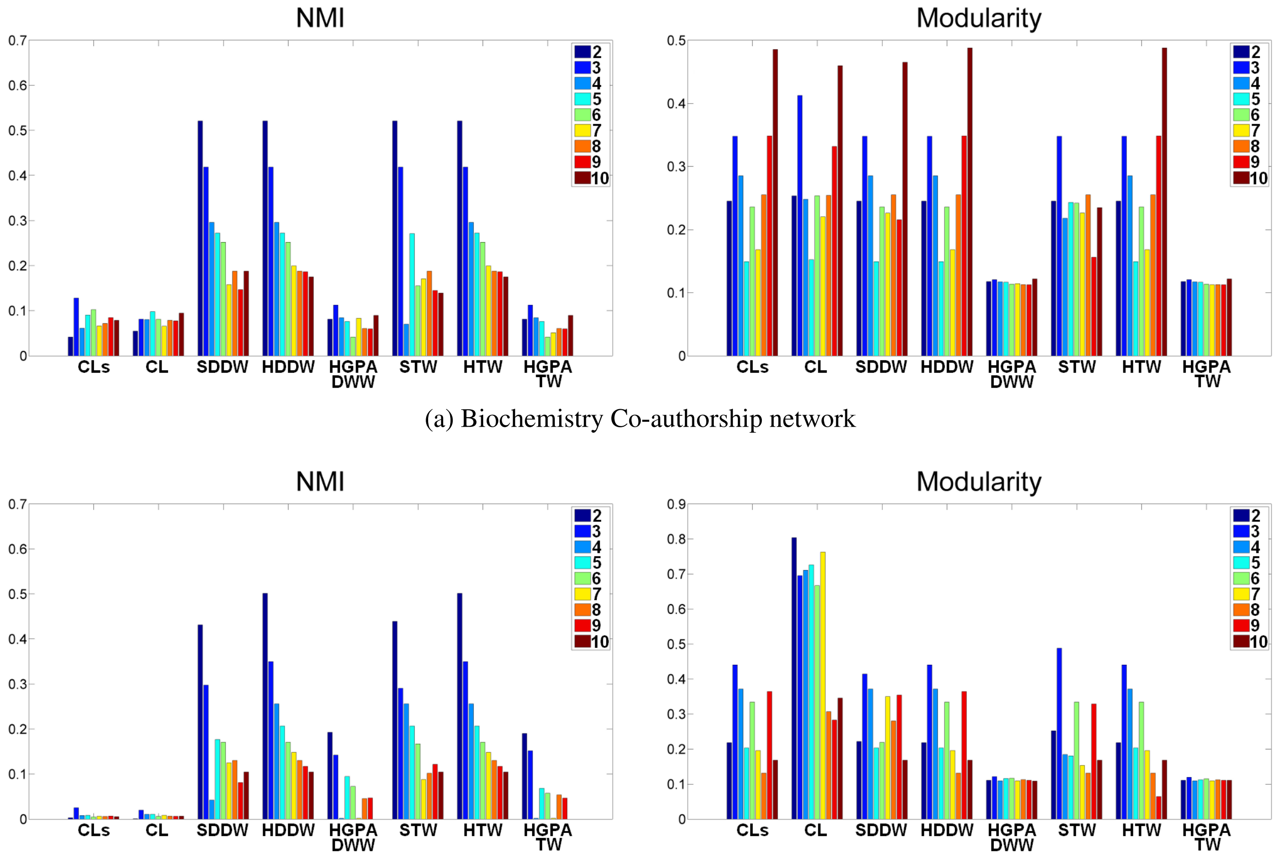
Regarding the NMI metric, our proposed spectral and hierarchical data weighting outperform the baseline CLs and CL for all timestamps in all used datasets. This shows that extracted roles by our method are more consistence over time and better shows the dynamic of the network. The DDW weighting function produces better results in comparison to the temporal weighting function (TW). This function assigns more weight to the historic data that has similar clustering structure to the current data. This basically reveals that some roles may exist in a network but not at consecutive time steps, hence the network structure at the current time is more similar to older times than just the previous snapshot of the network. In other words, if the topology of a network significantly changes over time, our method utilizing DDW function can still find the structural roles of nodes with high accuracy (modularity) and consistency (NMI) including the concept drift in the structure of the network. While the two baseline methods suffer from this drawback: the CL method uses the stacked dataset which is large and is likely to contain topological structure that is



(a) Autonomous systems network

(b) DBLP Co-authorship network

Figure 5.3: The performance of different methods in terms of NMI and modularity for the networks. CL and CLs: the two baseline methods, SDDW, STW, HDDW, HTW, HGPA DDW, HGPA TW: are respectively combination of spectral clustering, hierarchical clustering and HGPA clustering with data distribution (DDW) or temporal (TW) weighting functions. The color of bard in the figures represents the time step.



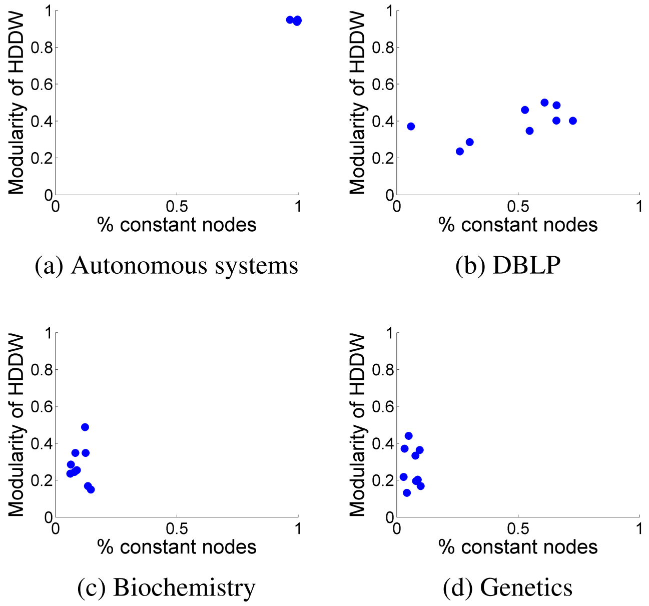
(a) Biochemistry Co-authorship network

(b) Genetics Co-authorship network

Figure 5.4: The performance of different methods in terms of NMI and modularity for the networks. CL and CLs: the two baseline methods, SDDW, STW, HDDW, HTW, HGPA DDW, HGPA TW: are respectively combination of spectral clustering, hierarchical clustering and HGPA clustering with data distribution (DDW) or temporal (TW) weighting functions. The color of bard in the figures represents the time step.

not valid for current snapshot; the CLs method only considers one time step data which may not be enough for clustering.

Out of three consensus clustering algorithms, HGPA has the worst performance for either weighting functions. The two other methods, spectral and hierarchical clustering are at the same level of quality. Further investigation of data revealed that the main reason for poor performance of HGPA was that it produces clusters with balanced sizes since it utilizes the HMETIS algorithm [KAKS97]. This algorithm partitions a graph with the constraint of producing even sized clusters. Whereas roles in a network are not equally distributed and some roles are at minority.



(a) Autonomous systems

(b) DBLP

(c) Biochemistry

(d) Genetics

Figure 5.5: The modularity of hierarchical ensemble clustering using DDW weighting function versus the percentage of constant nodes at a time for Autonomous systems network and Co-authorship networks of DBLP, Biochemistry and Genetics. The modularity drops when a large number of nodes join the network and no history of the temporal behavior of nodes is available.

Figure 5.4 also assesses the quality of the discovered roles by measuring the modularity, as explained before. The quality obtained by our method is higher or equal to the baseline methods except at the time steps that a large number of new nodes join the network. For the AS network, we have a constant number of nodes over time and at each time step the temporal behaviors of all nodes are available. As we can see from the results, our method outperforms the baselines when either spectral or hierarchical clustering is employed for ensemble clustering. Figure 5.5 shows the relation between the percentage of constant nodes and the modularity obtained by of our method at each time step. We can see that the accuracy drops off when the percentage of constant nodes in the network decreases. At some time steps for co-authorship networks, ERM has poor performance comparing to the baseline methods in terms of modularity. By examining the growth rate of the networks and the number of new nodes joining the networks at a time, it shows that the performance declines when a large number of new nodes join the network. This is reasonable, since ERM relies on the history of nodes to find their role as well as their current structure. Therefore for new nodes, where no historic data is available, out method cannot learn the roles with enough accuracy.

5.3 Conclusions

In this chapter, we presented an evolutionary clustering method for role extraction in networks. Our method finds the structural role of nodes regarding their current position in the network and their historic data. We utilize the ensemble clustering in our method where nodes at each time step are clustered by aggregating all the available partitionings of data in previous time steps. We use a weighting function to incorporate temporal smoothness into the evolutionary clustering method. We conducted an empirical evaluation using normalized mutual information (NMI) and modularity metrics to demonstrate the performance of our method in capturing evolutionary roles in networks. The modularity assesses how well roles fit to the current structure of the network and NMI metrics evaluate the closeness of current role to previous roles of nodes. The evaluation results on real world networks shows that spectral clustering and hierarchical clustering algorithms outperform HGPA method and have better performance than the baseline approaches as well. In addition, we defined DDW weighting function based on network structure to incorporate temporal aspect of network in role discovery. We showed that this function can better explore evolutionary roles in a network, comparing to a temporal weighting function.

We demonstrate how the method described here can be applied in a user behavior study scenario in chapter 6. We use evolutionary roles of users to infer the category of influential users in information cascades.

Chapter 6

Dynamic inference of structural roles in information cascades

What is the role mining application in social networks? Do structural roles of users reflect their social roles in a social network?

Users in online social networks get engaged in different social activities such as sharing and exchanging information. Information propagation models study how an idea spreads in social networks. These studies mainly consider users activity and their neighbors activity to model the process. In an information cascade, users behave differently: some are more active in terms of adopting new ideas, some cause blockage and others are more influential in spreading the ideas. Understanding social behavior of users is important in modeling the information propagation in many diverse phenomena, including adoption of new ideas, spread of infectious diseases, computer virus epidemics on the Internet, viral marketing campaigns, and information cascades in online social networks [AA05, EK10, MZL12, WSAL12].

Users behavior is essentially modeled based on the history of their activity and their friends' activity, that is, on the information flow in itself. The structural connectivity of the network has comparatively received much less attention. Regardless, it has been shown that network characteristics of users also affect their activity. For instance, Leman et al. show how users' influence is correlated to centrality measures [GL12].

In this chapter, we investigate precisely how the social status of users relates to their structural position in the network. Nodes at different topological positions, such as centers of stars, members of cliques and peripheral nodes may have different functions. The roles are defined using structural measurements of the node and its neighborhood. More specifically, we study information propagation of *stories* in social networks, and we concentrate on the effects of structural patterns on two different properties: level of *influence* and *blockage rate*. We categorize users into different roles in a social activity from these two points of view. User influence is related to the cascade size a user can cause, that is, the amount of other users that receive stories propagated by such cascade. Blockage rate amounts to the number of stories a user does not repost, normalized to the total number of received stories. We use network characteristics of users to classify them into social groups and try to find a correspondence between topological positions and social role.

Our end goal is to use structural roles to reveal social activity and to discover the essential connectivity principles behind social activities. For this purpose, we utilize the method proposed in chapter 5 to classify nodes and examine correlation between structural position and social activities.

6.1 Related work

Information propagation in social networks has been widely studied for a number of years from different aspects. We can divide past work in two major categories. The first one includes research works that study the process of influence spread and how the information propagates from one to another. The second category includes research studies that focus on characterizing users in order to find a set of nodes with maximal influence.

In the first category, several influence models have been proposed and studied, and the most popular ones are the linear threshold model (LT) and the independent cascade model (IC), by Kempe et al. [KKT03]. These models study spread of influence through social networks, where the influence probabilities between users are predefined. Saito et al. [SNK08] predict the influence probabilities in independent cascade models of propagation by maximum likelihood estimation and Goyal et al. [GBL10] study the probabilities in the threshold model by counting the number of correlated social actions. They both consider the temporal nature of users' influence.

The second category of research works in this field, measure users' influence by some structural models of influence like PageRank and in-degree centrality in the network [KLPM10], number of followers, mentions, retweets [LKPM10, CHBG10] or the size of the information cascades [BHMW11]. Earlier studies of social influence and propagation, showed that the most influential bloggers were not necessarily the most active [ALTY08]. Temporal information has been used in modeling influence using the influence-passivity score [RGAH11]. An important aspect of information dissemination is the study of parameters that stop the contagion. Steeg et al. showed that many of cascades grow slower than expected and do not reach “epidemic” proportions [VSGL11]. Their study on Digg data showed that multiple exposures to the same information does not affect the probability of voting. The same phenomena is seen on Flickr data where the photos are not spread in a quick and viral fashion throughout the social network [CMG09b]. Although the structure of the Flickr social network holds small-world properties, which in theory says a piece of information will spread quickly and widely through social links, photos on Flickr are spread with delay [CBAG12]. This study concludes that propagation is not only due to activity of users but also due to information availability at the time of users' activity.

Using multiple sources of information may raise the complexity of the analysis, but also brings more resolution to the problem. Tang et al. [TSWY09] leverages another source of information for finding topic-specific influence. They use topic distribution of users in conjunction with a social network of users to build a factor graph model, and propose a topical affinity propagation on the factor graph to automatically identify the topic-specific social influence. Zhou and Liu [ZL13] integrated three sources of information to derive the influence group of users. They defined a new similarity matrix between users based on three sources of information including a social network of users, activity networks and influence networks. They proposed a clustering algorithm based on k-means which divides users into homogeneous groups regarding the derived similarity matrix. They combined social influence based similarity between each pair of users by unifying the self-similarity and multiple co-influence similarity scores through a weight function with an iterative update method.

All of these papers use both the activity log of users and their social network to characterize the influence process. However, in this thesis, we use only the topological properties of users to categorize their role in how the information is spread.

6.2 Relation between network topology and social activity

In this section we analyze the role of users in information cascades and how network characteristics of individual nodes affect their social activities. We quantitatively study the relation between a number of structural properties of users in a network and two aspects of

Table 6.1: Summary of datasets.

Data	#Users	#Objects (story/photo)	#Links	Network time interval	Time granularity
Digg	71,367	3,553	1,731,658	5 years	three months
Flickr	914,400	4,000	18,595,048	2 years	one month

information cascades: user influence and user blockage.

Throughout this chapter, we will be using two different datasets, coming from two well known and established internet communities: *Digg*[1]and *Flickr*[2], the same datasets that were introduced section 4.3.1. Both datasets include a static social network including social relationships between users and a dynamic evolving network describing information propagation. Table 6.1 summarizes the statistics of the data we used.

6.2.1 User Influence

In order to study how the topology of networks relates to the influence level of users in an information cascade we measure the influence score of users for information propagation in networks. We assign social roles to users using the influence score in an information cascade. There are two different definitions for the empirical influence of users: 1) size of the cascade initiated by a user [BHMW11]; 2) number of votes a user's stories receives from his fans [GL10]. These definitions are limited to submitters but in a cascade other users also play important roles in spreading the information and have some levels of influence in the cascade. Hence, we adopt the second definition for all users as influence score:

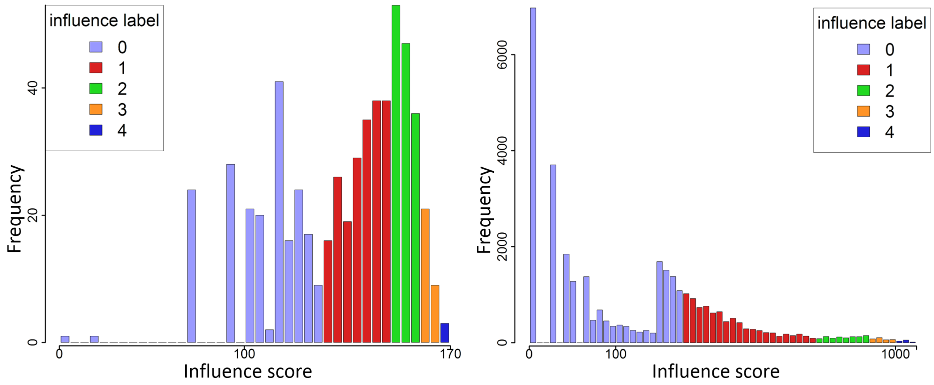
$$Influence(u_i) = \sum_j \frac{votes_f(s_j)}{|posts(u_i)|}, s_j \in posts(u_i) \tag{6.1}$$

where $votes_f(s_j)$ is the number of votes story s_j receives from fans of user u_i after user u_i has voted, and $posts(u_i)$ is the set of all stories submitted or voted by user u_i.

For example, in Digg data when a user submits a story it becomes visible to his fans. Some of his fans may like the story and vote for it, making the story visible to their fans as well, and this process goes on. All users are important in spreading the information but

[1] http://www.isi.edu/ lerman/downloads/digg2009.html

[2] http://socialnetworks.mpi-sws.org/datasets.html



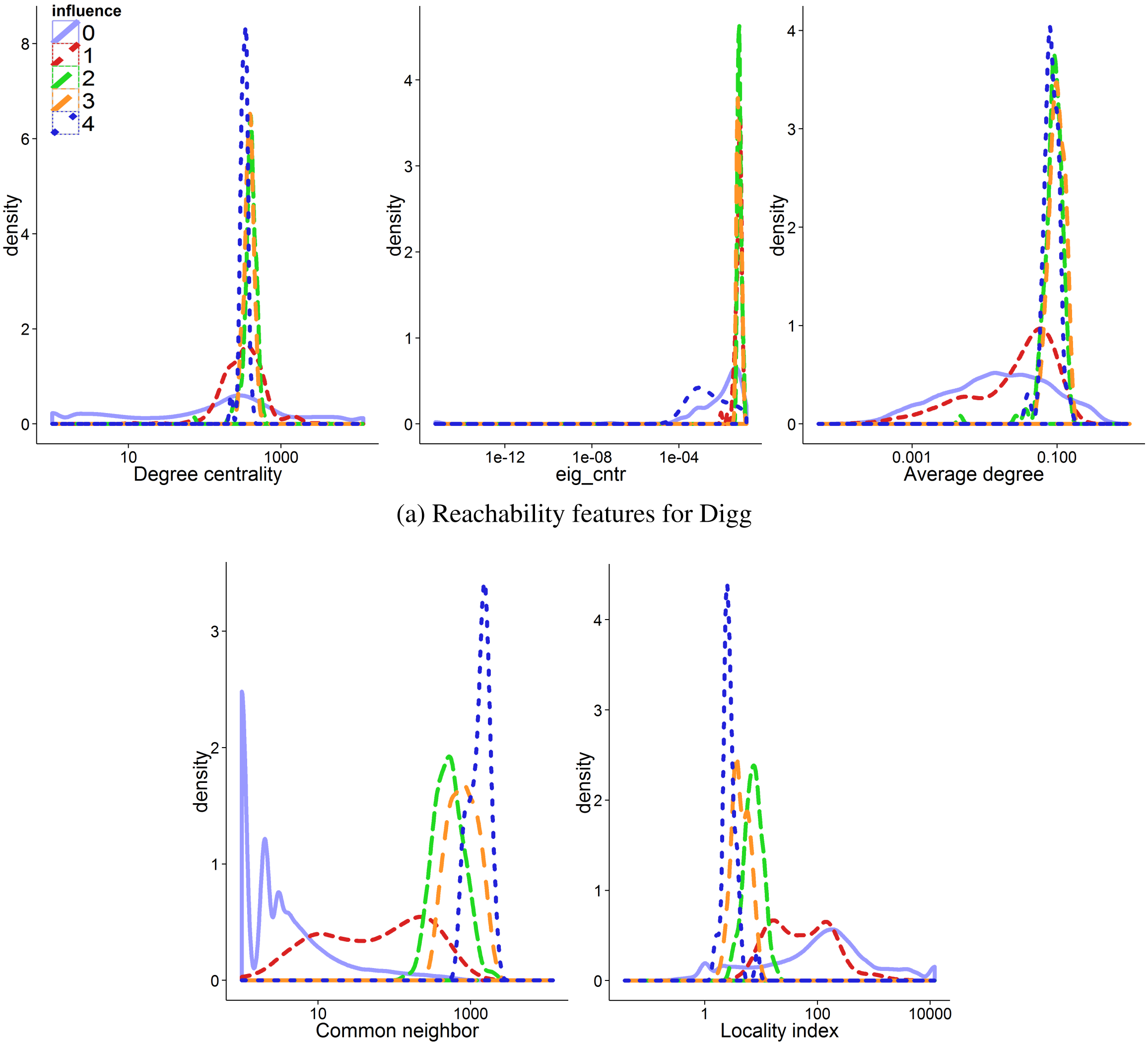
(a) Digg

(b) Flickr

Figure 6.1: The histogram of influence score in a semi- logarithmic scale for users in Digg and Flickr data.

at different levels. Figure 6.1 shows the distribution of influence scores for users in Digg and Flickr social networks, the plot is shown in logarithmic scale as the value of influence is long tailed. We categorize users into different groups regarding their influence score. We use equal width discretization to factorize the influence value and classify users in a network into five groups from non-influential, to highly influential. In the figure groups are highlighted with different colors. The influence models mentioned in section 6.1 are basically built on the individual users features and do not take into account the neighborhood properties. In this chapter we study the effect of neighborhood structure on users' influence. We examine the correlation of structural features on the influence of users regarding reachability and commitments of users.

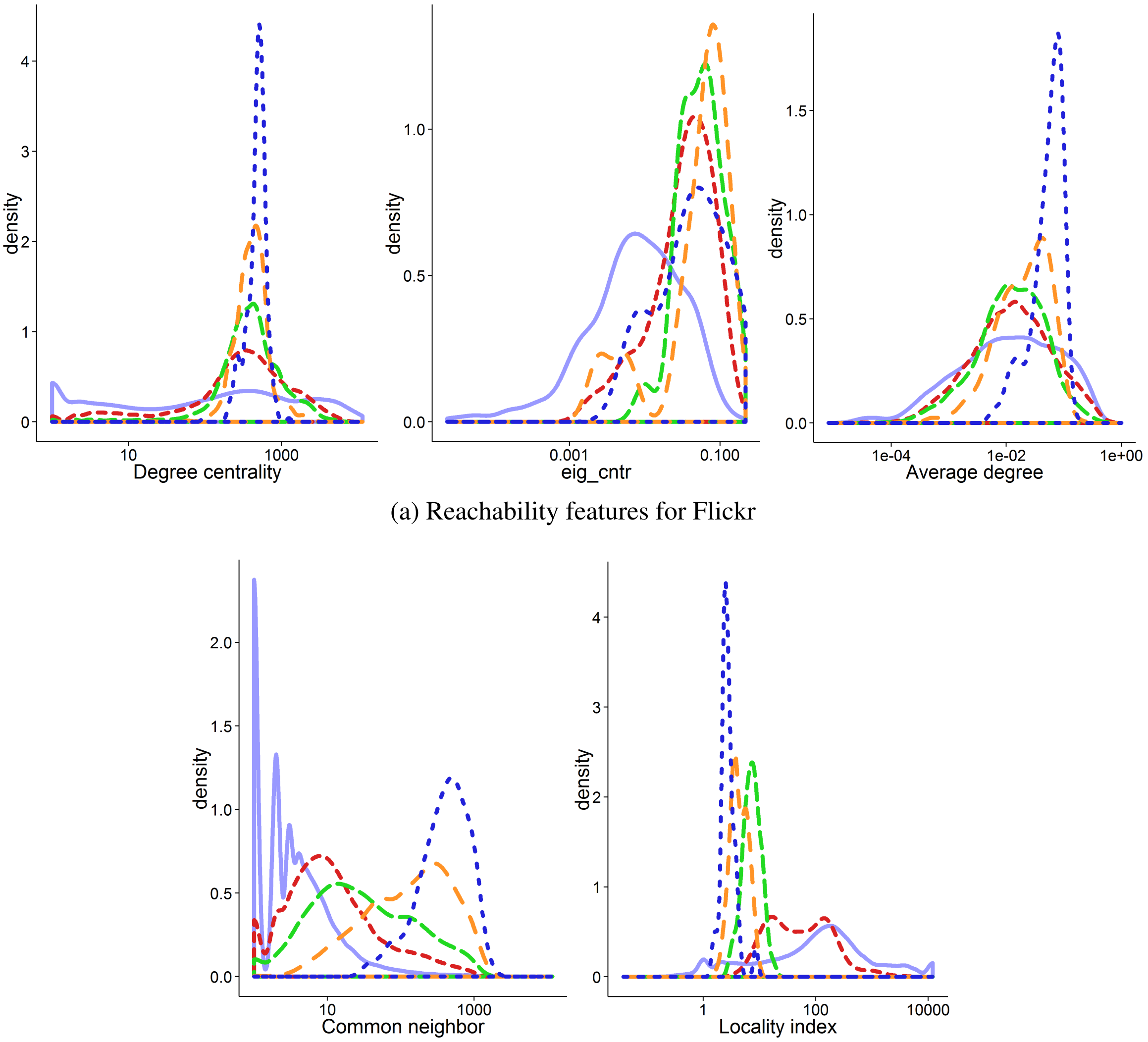
Reachability of users is important for spreading information and many of the influence models are based on this property. We quantify reachability of a user in a network by using "degree centrality" defined as the number of users directly connected to a user. We also study the degree distribution in the neighborhood of a user by measuring the "average degree in neighborhood". This property represents the "2-hop" reach of individuals in the network. Out of three centrality measures of betweenness, closeness, and eigenvector, we have selected eigenvector which had higher distinguishing power. We examine the eigenvector centrality [Bon07] of users, which rank users regarding their importance in the network. This centrality metric acts similarly to degree centrality. However, it gives higher score to the nodes which are themselves connected to high score nodes. In other words, the quality of neighbors of a node is accounted in eigenvector centrality. Figures 6.2 and 6.3 show the distinguishing power of these three reachability measurements for users in Digg and Flicker networks. We can see that the distribution on all five influence groups



(a) Reachability features for Digg

(b) Commitment features for Digg

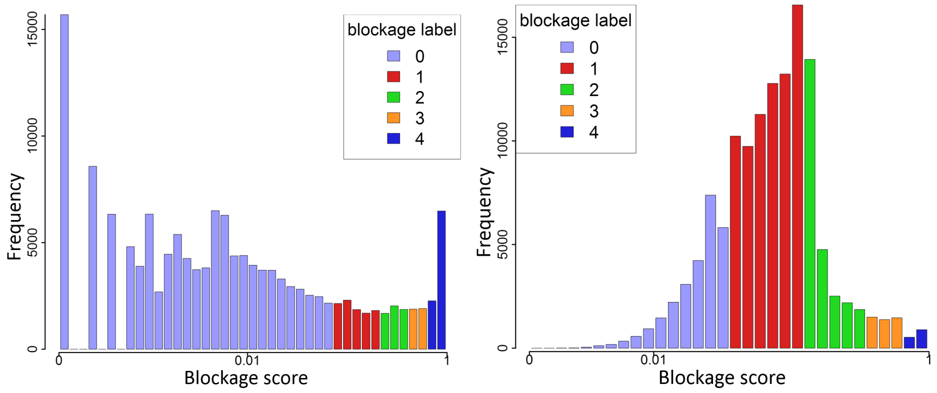
Figure 6.2: Correlation between social roles of users in an information cascade network and their network characteristics including degree centrality, average degree, eigenvector centrality, common neighbor and locality index at different levels of influence in Digg social network.



(a) Reachability features for Flickr

(b) Commitment features for Flickr

Figure 6.3: Correlation between social roles of users in an information cascade network and their network characteristics including degree centrality, average degree, eigenvector centrality, common neighbor and locality index at different levels of influence in Flickr social network.



(a) Digg

(b) Flickr

Figure 6.4: The histogram of blockage rate in logarithmic scale for users in Digg and Flickr data.

for eigenvector centrality is more easily distinguishable. Furthermore, the average degree in neighborhood has more discrimination power than degree centrality.

Commitment of users to their neighborhood is another important feature that we study. It essentially shows how well a user is connected to his neighborhood comparing to the whole network [Gra85]. We study the effect of this feature on influence of users by measuring two structural properties: "locality index" (LOC) and "common neighbors" (CN) . LOC is the ratio of the number of connections between neighbors and the rest of the network to the number of connections between neighbors of a user. CN is the number of common neighbors between a neighbor of a user and his neighbors' connections. Figures 6.2 illustrates the distribution of commitments properties We can see a peak in the LOC distribution graph, belonging to the fourth group of influence category, which includes users with very high influence score. These users all have a very low locality index, varying in short range in the other word the variance of LOC in this group is low , which means they are located in a dense neighborhood and their neighbors are more connected to themselves than to the rest of network. The plot of common neighbors confirms these results, as we see that users in this group share many neighbors. We also observe that the probability distribution of Loc distinctly changes from a influence group to another. Generally, we can see that commitment properties can better distinguish users at different influence level, when compared to reachability properties.

6.2.2 Cascade Blockage

Many of the cascades grow far slower than expected from their initial spread and fail to reach epidemic proportions [LGS12]. The network structure somehow limits the growth of cascades. In this section we study the role of users in stopping a cascade. We define the blockage rate as the probability of not voting for a story if at least one of the users' friends has voted for it. We estimate this probability as the fraction of stories visible to a user and that he did not vote for. We define a new measurement to formulate it as:

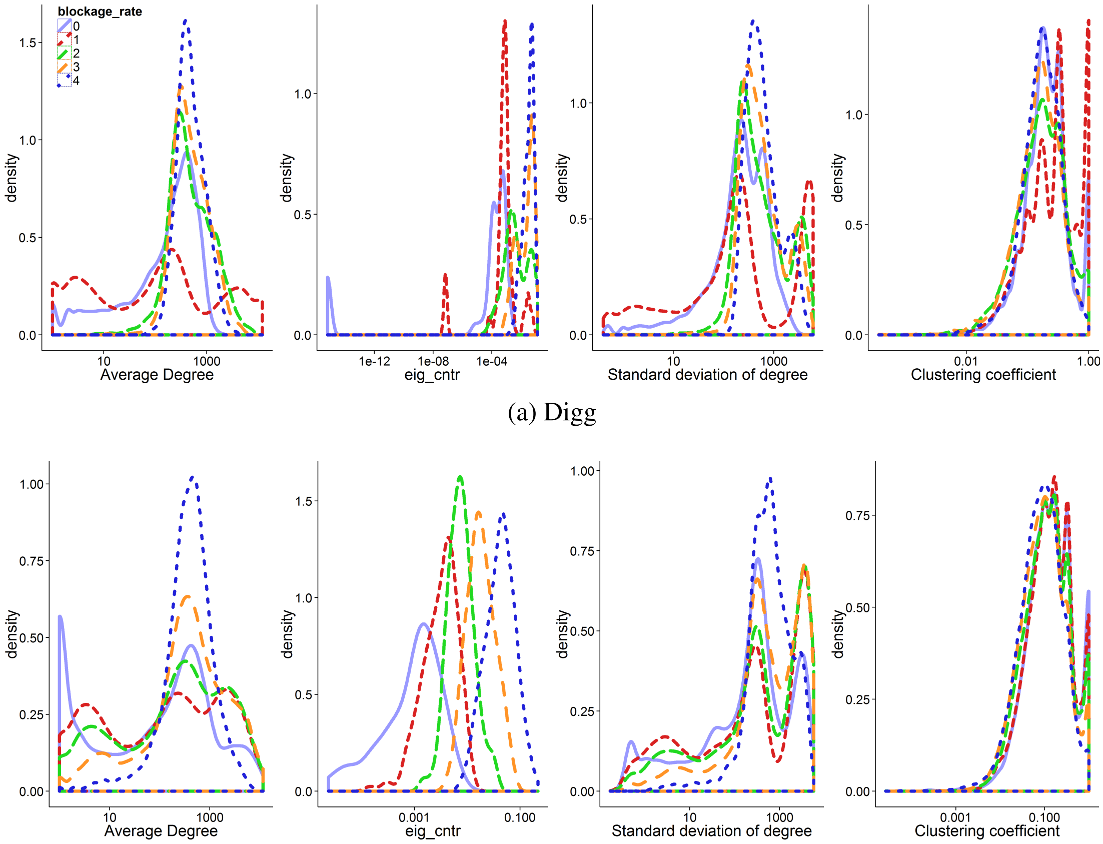
$$Blockage(u_i) = \frac{\sum b(s_j)}{|received(u_i)|}, s_j \in received(u_i) \tag{6.2}$$

where $received(u_i)$ is the set of stories visible to user u_i and $b(s_j)$ is 0 if u_i repost s_j and 1 otherwise.

We categorize users into different groups based on the blockage rate using equal width discretization method, similarly to what we did with user influence. Figure 6.4 shows the distribution of blockage rate of users in Digg and Flickr data in a logarithmic scale. Based on this distribution we have five groups of users, shown in different colors from non-blockers to blockers. Non-blockers are users with very low blockage score. We investigate the correlation of blockage rate of users against three structural properties of users in a network.

Triadic closure is an important property that represents the triangular structure of a network [Gra73]. The local triangular structures in networks is a fundamental feature that causes spread of information in networks [IM09]. To incorporate it in our method we use the local clustering coefficient of each user [KW06, GWL11]. This measures the number of triangles (cliques of size 3) a user i is involved in, normalized by the number of triplets of connected nodes (not necessarily cliques) that include the same user i. The clustering structure of networks causes multiple exposures to a story and this may limit the spread of information [LGS12]. Figure 6.5 shows the distribution of clustering coefficient of users at different levels of blockage rate in two social networks of Digg and Flickr.

In addition to the triadic closure we examine the relation of blockage rate with other network characteristics. We assess the effect of neighborhood cohesion on the behavior of users. To quantitatively measure the cohesion in the neighborhood of a user, we use standard deviation and euclidean mean of degree of connected users. The cohesion of the neighborhood of a user appears to be more effective on determining the user's voting behavior. As we can see from Figure 6.5, distributions of average degree and standard deviation of degree for users



(a) Digg

(b) Flickr

Figure 6.5: Correlation between social roles of users in an information cascade and their network characteristics including average degree, eigenvector centrality, standard deviation of degree in neighborhood and clustering coefficient in Digg and Flickr social networks.

Table 6.2: Correlation between structural properties of users and their social activity in Digg network.

Influence score	Average degree	Degree centrality	Eigenvector centrality	Common neighbor	Locality index
	0.067	0.061	0.26	0.13	0.095

Blockage rate	Average degree	Eigenvector centrality	Standard deviation of degree	Clustering coefficient
	0.063	0.41	0.081	0.076

with high blockage rate are more shifted to right and are centered around an higher average comparing to the groups with lower blockage rate.

We also examine how the centrality of users affects their blocking behavior. We use eigenvector centrality of users and we can see that eigenvector centrality of users at different blockage rates has different ranges. This is more obvious in the Flickr dataset, as we can see in Figure 6.5.

In summary, network characteristics of users affect their social activities and can be used to categorize users into different social roles. The correlation analysis among cascade properties (influence score and blockage score) and structural properties of users are depicted in Table 6.2. As we can see the Pearson correlation values are not strong nor negligible. We note that most of the structural properties are weakly correlated and independently can not infer social roles of users. In the following sections we show how we can distinguish users in terms of their social activity only by using the ensemble of their structure properties in the social network.

6.3 Structural roles vs Social roles

In this section we show how the evolutionary role assignment method proposed (ERM) in chapter 5 can be used for categorizing social roles of users in an information cascade. We applied ERM on two social networks from the Flickr and Digg datasets, described in section 4.3.1. We evaluate the efficacy of the structural roles from two angles: accuracy in categorizing users using F-score and temporal consistency of the extracted roles using normalized mutual information. We use the influence score and blockage rate of users, as measured in sections 6.2.1 and 6.2.2, to define two sets of class labels as ground truth to compare our results against. These class labels are called “influence” and “blockage” and are respectively derived from equal width discretization of measured values of influence score and blockage rate where the number of intervals is 5.

6.3.1 Number of roles

We determine the number of roles by measuring the F-score for different cluster sizes. We apply our method on the datasets for different cluster sizes from 5 to 25 and measure the F-score twice for each dataset: 1) F-score of results against influence categories as true labels; 2) F-score with blockage categories as the true labels. Hence we have 2 sets of results per dataset as shown in Figure 6.6 for different number of roles. As we see, the best result (maximum F-score) is derived for cluster size 6 for Digg data and cluster size 8 for Flickr data. The F-score of p on r, denoted as $F(p, r)$, is the harmonic mean of precision and recall rates. For a predicted role p, we compute its F-score on each r in the actual roles of R and define the maximal obtained as p's F-score on R, i.e., $F(p, R) = \underset{r \in R}{max} F(p, r)$. The final F-score of the predicted roles P on the actual roles R is then calculated as the weighted (by role size) average of each predicted role's F-score:

$$F(P, R) = \sum_{p \in P} \frac{|p|}{|V|} F(p, R) \tag{6.3}$$

For the predicted groups of p with reference to actual roles r (which are sets of nodes), the precision rate is defined as $\frac{|p \cap r|}{|p|}$ and the recall rate is defined as $\frac{|p \cap r|}{|r|}$. This effectively penalizes the predicted clustering that is not well aligned with the ground truth, and we use it as the quality measure of all methods on all datasets.

We varied the number of clusters from 5 to 25. If we assume that influence score and blockage rate are 100 percent correlated then there will be only 5 social groups, corresponding to the previously defined 5 equal-width groups. By contrast, if they are completely not correlated, there will be $5 \times 5 = 25$ groups. In practice, we found that the actual correlation between these two scores of users in Digg and Flickr social networks is respectively 0.13 and 0.24.

In the context of this thesis, we us F-score to evaluate the optimal number of clusters since we already have the desired labels of the users. In other applications, any method such as AIC [Aka98] can be incorporated in our framework to determine the appropriate number of clusters.

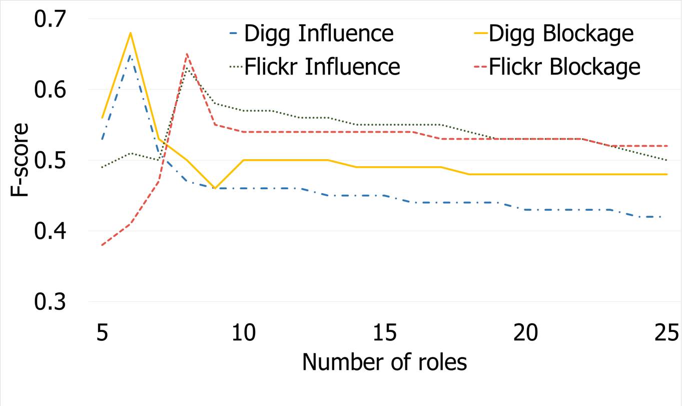


Figure 6.6: Performance of derived roles by proposed method for different number of social roles in terms of F-score. The F-score is measured against two true class labels: influence and blockage rate.

6.3.2 Baseline methods

To show the effectiveness of our method, we compare the results of the proposed methodology against four baseline approaches. Since our method considers both the temporal behavior of users and their local structural properties, we use the following approaches to evaluate the performance of our method and study the effect of the clustering algorithm and the historical information:

- **Single time**: This method studies the effect of historical data where only the local properties of users at the current time step are considered and the temporal behavior is discarded. In this method, we use k-means and spectral clustering to derive the social roles in the current snapshot of the network.
- **Stacked**: In this method the temporal data is incorporated in the clustering by stacking all structural properties of users at each time step up to current time. The clusters are derived by applying a clustering algorithm on the stacked data, using k-means and spectral algorithms. This is the general approach in evolutionary clustering where all data is available. In addition, previous studies of dynamic role discovery employ this strategy [CSR11, RGNH12].
- **RolX**: We also compare our method against the method proposed by Henderson et al. [HGER$^+$12]. They use a set of structural features of nodes in networks and extract their role by applying matrix factorization method to cluster nodes where each cluster represents a role. We extract the same feature vector as RolX, including local features, neighbor features and recursive features for the current snapshot of network.

- **RolX-stacked**: This method finds clusters of users by applying RolX method on the stacked matrix of features where structural properties of users over time are aggregated into one matrix.

6.4 Temporal consistency of structural roles of users

The role of a user is not independent from its temporal behavior in the network, i.e., history of adding or removing links also affects its influence on the way information is propagated, as is depicted in Figure 6.7. In particular, we examine the correlation between influence and the rate at which a user builds new connections over time. We define an user's degree growth rate as:

$$Growth\,rate(u_i, t) = \frac{degree^t(u_i) - degree^1(u_i)}{t} \tag{6.4}$$

where t is the age of user. As one can see from Figure 6.7, the influence score of a user i is correlated to his temporal behavior. $Growthrate(u_i, t)$ is the difference between the initial degree of a user and its latest degree, normalized by its age.

Users with a large growth rate attract more connections per time step. Influential users have large growth rates, meaning that they attract new connections faster. Social role mining is a dynamic process and here we evaluate the quality of the obtained social roles in terms of the temporal smoothness of extracted roles. We use the normalized mutual information (NMI) [SG03] measure to quantify the amount of mutual information shared between roles of users at previous time steps and current time.

In Figures. 6.8 the performance of different weighting functions and algorithms on each dataset is compared against the baseline approaches. Each bar in this figure represents the average normalized mutual information (ANMI) between the set of r clusterings over time, R and the clustering at current time λ^T. In other words, we find the clustering of users for each timestamp t, then we compute the NMI relative to the clustering t and T, and we average the NMI values relative to all t. The performance of single time and stacked approaches for both clustering algorithms (k-means and spectral) are very similar. Thus, we only demonstrate the results of spectral clustering, in order to have the same base clustering algorithm for all methods, including ours, for a better and fairer comparison. The results of both datasets have a very similar pattern. We can see that our method outperforms the baseline approaches if the ensemble clustering algorithm is either spectral or agglomerative hierarchical clustering. These methods, spectral and agglomerative hierarchical clustering

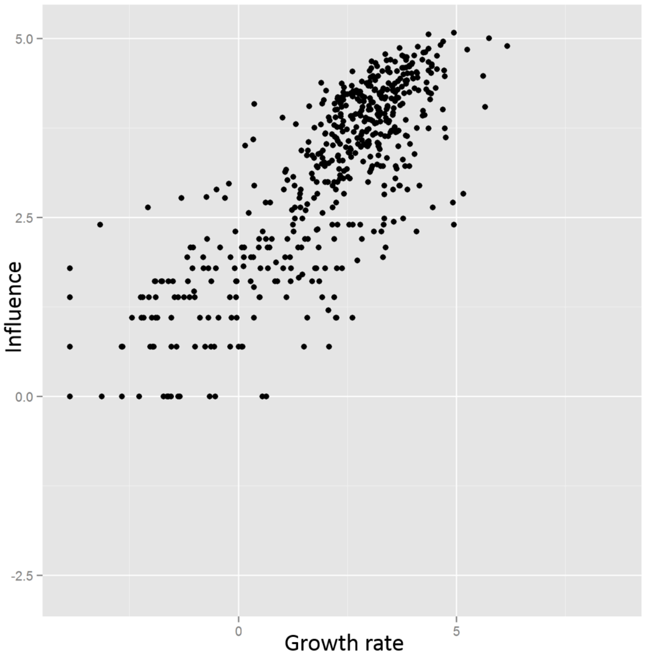
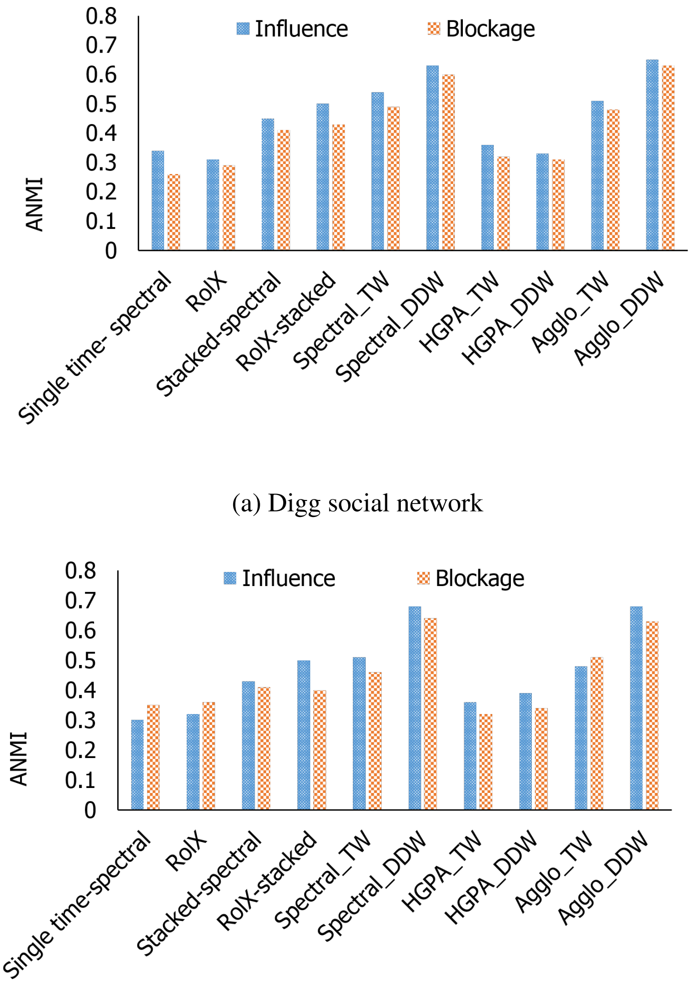


Figure 6.7: The influence score and degree growth rate of users in Digg dataset. One pair in the plot shows influence of a user and the rate at which one user has built new connections over time.

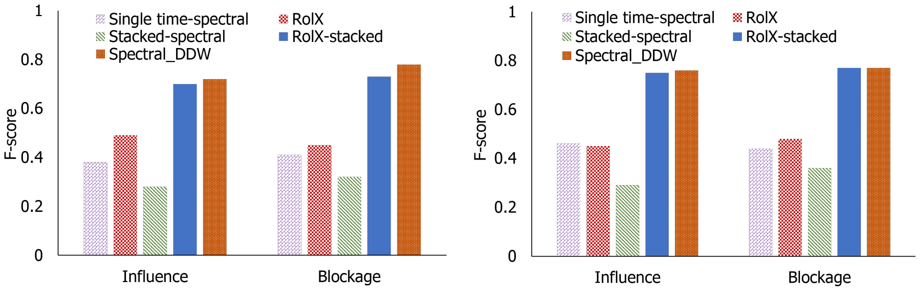
are at the same level of quality. This shows that the roles extracted by our method are more consistent over time and better show the dynamic of the network. The DDW weighting function produces better results in comparison to the temporal weighting function (TW). This function assigns more weight to the historic data that has similar clustering structure to the current data. This suggests that social behavior of users is not monotone over time. Hence, the network structure at the current time is more similar to older times than just the previous snapshot of the network. In other words, if the topology of a network significantly changes over time, our method using DDW function can still find the structural roles of nodes with high accuracy (F-score) and consistency (NMI) including the concept drift in the structure of the network. By contrast, the two baseline methods suffer from this drawback: the stacked method uses the stacked dataset which is large and is likely to contain topological structure that is not valid for current snapshot; the single time method only considers one time step data which may not be enough for clustering. RolX also presents the same issue.



(a) Digg social network

(b) Flickr social network

Figure 6.8: Average normalized mutual information (ANMI) score for different approaches and proposed method



(a) Digg social network

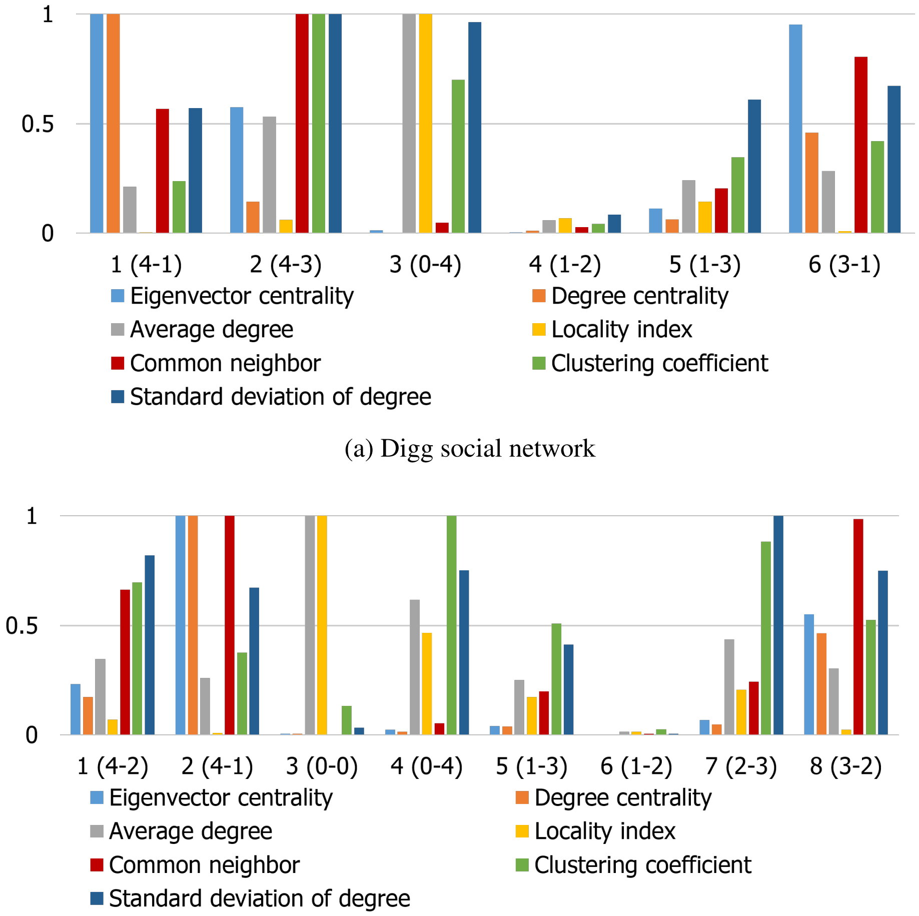
(b) Flickr social network

Figure 6.9: F-score performance of different approaches on dataset

6.5 Performance in social role of users

In this section we investigate how well the role of users correlate to their social influence and their function in information dissemination. We use the empirical influence rate and blockage rate of a user, as defined in sections 6.2.1 and 6.2.2, to label users to groups as the actual role of users in a cascade. We run our algorithm for Digg data for 6 roles and for Flickr data for 8 roles as we saw in Figure 6.6. We compare the predicted social roles of users from our method with their actual roles (influence level and blockage rate) using F-score. Figure 6.9 shows the experiment results on Digg and Flickr data. In this figure we compare the performance of our methodology and the baseline approaches. For the sake of more clarity we only compare our methodology using spectral clustering algorithm and the weighting function is data distribution since this configuration has better NMI performance over HGPA and relatively similar performance to agglomerative hierarchical clustering.

From the figures, we can see that the Spectral-DDW outperforms the baseline methods on F-score measures for both datasets of Digg and Flickr. Comparing to the single time approach, Spectral-DDW improves the performance by 0.3. This is in accordance with our observation that temporal behavior of users is important in categorizing users social roles. This also suggests that the Spectral-DDW method is capable of incorporating history of users to infer their social roles. The Spectral-DDW method produces an improvement over staked method as well. We can also observe that the performance of the stacked method is lower than a single time approach. Since this method uses the entire history of users, it is biased toward past. Hence, it does not reflect the current social roles of users accurately. Our proposed method outperformed RolX for both datasets. This can be due to the impact of temporal behavior of users in their social roles since the history of users is not considered in RolX. We also note that the Spectral-DDW method improves the F-score for the category of social roles on both the influence and blockage levels. This means that the Spectral-DDW proposed model is capable of categorizing users to their social roles in an information cascade. All these results hold for both datasets of Digg an Flickr, which shows the robustness and consistency of our method over different datasets. We can see that the results of RolX-stacked method and our proposed method are comparable. These experiments show that the quality of results by our method is not only due to incorporating temporal data but also is due to the method used for aggregating the history of users and their current status. In the stacked baseline approach, history is used and the base clustering is the same as in our method. However, the quality of results is lower than the quality of the roles discovered by ERM.



(a) Digg social network

(b) Flickr social network

Figure 6.10: Average local properties of users in predicted roles. The numbers on the x-axis shows the predicted roles and corresponding social category in the form of (influence - blockage).

Figure 6.10 shows the average statistics for predicted roles in Digg and Flickr dataset. The x-axis shows the role number and each column represents the average value of local properties of users associated with predicted roles. For a better visualization, we normalized all the values to [0,1], so that we have the same range of value for all properties. We also report the category of users regarding their influence score and blockage rate for each predicted role. On the x-axis, the the numbers in the parentheses shows the category in form of (influence category - blockage category). These categories are the ones that have highest F-score with the predicted role. For both datasets we can see very similar patterns for grouping of users. Some of the roles are exactly the same in both dataset such as role 3 in Digg dataset and role 4 in Flickr dataset and the other have corresponding from one dataset to other such as role 2 in Digg and 1 in Flickr. We group the resulted roles based on their influence-role category where the numerical label of zero to four are translated to very low to very high in each category as follows:

- *Role A (very low influence, very high blockage)*: The structural properties of this role are also very similar in both datasets. We can see that users with this role are located in a neighborhood with low cohesion as the variance of degree is high and reachability of users is low as their centrality measures are low and they are not committed to their neighbors which also shows that level of trust in their neighborhood is low as they do not have much friends in common [EK12]. Considering that these users have low influence and block many stories, this feature profile very well represents this role. This role corresponds to group 3 in Digg and group 4 in Flickr.

- *Role B (very high influence, low blockage)*: Users with this role are very reachable and are located in a dense neighborhood with medium cohesion. As one can see from Figure 6.10, commitment of users to their neighborhood is very high. Hence users are very influential and do propagate most of the stories they receive from their neighbors. This role includes group 1 from Digg and 2 from Flickr.

- *Role C (very low influence, very low blockage)*: These users are not influential although they do propagate most of the received stories. This is can be interpreted as this role belonging to users whose neighborhood is not well connected, but globally they are well connected as their locality index is very high. In addition, one can see that these users are connected to high degree users as their reachability features are low except for average degree. This role is only observed in Flickr dataset in group 3.

- *Role D (low-mid influence, high blockage)* Users in this group have low reachability but are connected to high degree users. Their neighborhood is nor as dense as users in role 2 and they are more committed to their neighbors than users in role 1. This role corresponds to group 5 in Digg and groups 5 and 7 in Flickr.

- *Role E (low influence, mid blockage)* Users in this role are not reachable nor committed to their neighbors. This structural position prevent them from being influential. In addition, their neighborhood is very coherent which means they are connected to users with similar degree which is low. This can explain why they are not very active users as they do not propagate many stories. This role corresponds to group 4 in Digg and group 6 in Flickr.
- *Role F (very high influence, mid-high blockage)* Users in this role are very influential but do not propagate most of the received stories. Structurally, they are very similar to role 2 except that they are not as reachable as users in role 2. This role corresponds to group 2 in Digg and group 1 in Flickr.
- *Role G (high influence, low-mid blockage)* What makes this group different from role 1 and role 6 is that they have high degree but are mostly connected to users of low degree. This role corresponds to group 6 in Digg and group 8 in Flickr.

6.6 Role generalization

In this section we examine the generalization power and the predictive ability of the discovered roles by applying our method for a classification task. We use the derived clusterings in the training set to classify the unlabeled users in the test set. We also use the cluster membership matrix of the users, which contains their distance to the centroids of the derived clusters. We exploit this matrix as a feature set with logistic regression to predict the role of users.

We also compare the predictive power of the discovered roles by our method (ERM) to the structural properties that ERM uses as input. We configure ERM such that the spectral clustering is used for ensemble clustering and data distribution weighting (DDW) function is used for incorporating temporal behavior in the role mining framework . We build a classifier (structReg) using logistic regression on the structural properties as input, instead of using the role membership matrix obtained from ERM.

We divide the users in the network into 10 parts and, for each experiment, we select one section as test set and run ERM on the other 9 sections as a training set to find the roles and build the cluster membership matrix (9-fold matrix). We build the classifier on the 9-fold matrix

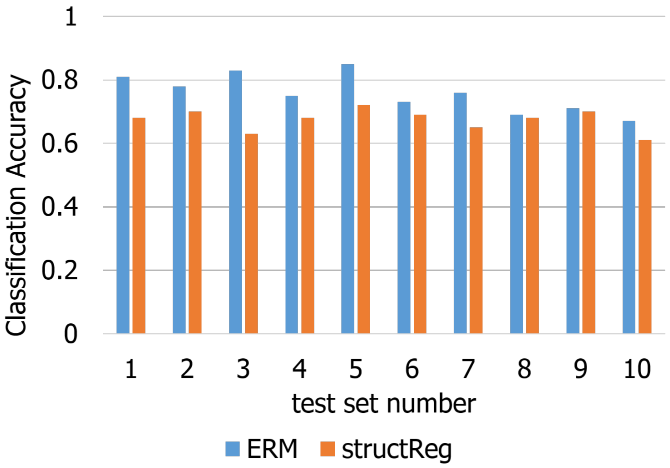


Figure 6.11: The classification accuracy for Digg dataset over 10 different test sets.

then predict the label of users in the test set (10th part of data which is not used by ERM). Here we use the labels of users derived from influence score categorization in section 6.2 as the class label. The results of this experiment on Digg dataset is shown in Figure 6.11. The roles produced by ERM are capable to classify the users more accurately than structReg in the test set. (ERM=78%, structReg=65% average accuracy, p-value=0.012 [3]). These results also show that ERM is capable to generalize more effectively than structReg, as we can see that the discovered roles on a part of a dataset can better classify users in the rest of the dataset.

6.7 Conclusion

In this chapter we study online social networks and we try to infer social roles of users in information propagation based solely on their structural properties, and not on the information cascade itself. We divided users into five social roles under two categories of influence and blockage rate, which are two important characteristics of users regarding information propagation. We demonstrated how our novel dynamic role mining method, explained in chapter 4 can help user behavior modeling in information cascade process. The experimental results, using two real social network datasets, show that the proposed model greatly outperforms a number of baseline models and is able to effectively infer roles of users in an information cascade scenario. In our experiment, we have shown that the quality of the results obtained by our method is not only due to incorporating temporal data, but also due to the method used for aggregating the history of users and their current status.

[3] The p-values are obtained from a one-tailed paired t-test.

In this study, we also explore the relation between users activity and their structural position. Among structural properties, we find that user commitment in the neighborhood has more impact on the influence score of users. In addition, neighborhood cohesion has a more additive effect on users' behavior in terms of blocking a cascade, and triadic closure is also useful. Our experiments show promising results in terms of correlation between user activities and their temporal structural properties and our model provides a step towards modeling an information cascade independently from the network of diffusion.

Part II

Motif mining

Chapter 7

Background

Given a large graph with weights over the edges, how can we find outstanding substructures? What function can measure the significance of a pattern regarding the weights other than the frequency?

A large body of knowledge has been developed for pattern mining in networks [YH02b, HWP03b, Ino04, KK05, KK04], since it has applications in a broad range of fields such as sociology [BGD$^+$06], biology [KGS04] or transportation networks [JVB$^+$05], where entities are modeled as nodes that are connected if they have interactions or are related. However, most of the developed methods are dedicated to unweighted networks, without taking into account the strength or capacity of the connections.

In this chapter, we review the definition of motifs and methods for motif mining in binary (unweighted) networks. We also examine how the motif concept and methods are extended to weighted networks.

7.1 Motif definition

A pattern in a network is normally defined as a subgraph which is very frequent or infrequent (in case of anomalies). A specific form of patterns are called motifs, which can be thought of as small subgraphs that appear in a network at significantly higher frequencies than what would be expected in similar randomized networks [MSOI$^+$02]. This type of patterns can significantly help in characterizing the networks, since they are not frequent only by chance, and therefore highlight the specific structural properties of the networks. That is why motifs are also known as the building blocks of networks, and it has been demonstrated that they can have functional significance in networks such as transcriptional regulation [SOMMA02] or protein-protein interaction [AA04].

In the next two sections we give a more detailed view on the motif concept both on unweighted and weighted networks.

7.1.1 Binary networks

The main element in definition of a motif is the notion of significance associated with each subgraph. In other words, a motif is a subgraph that has a significance score rather than frequency. Originally, Milo et al. [MSOI^{+}02] defined statistical significance of a subgraph in form of $z - score$ where frequency of the subgraph in the original network is compared to the frequency of the subgraph in random similar networks. The intuition behind this definition is to make sure that the intrinsic global and local properties of the network do not determine the motif appearance and that the motif is indeed specific to this particular network. Therefore, a series of networks similar to the original one are randomly generated by maintaining all single-node properties, namely the in and out degrees. Figure 7.1 exemplifies this concept and, as we can see, the number of incoming and outgoing edges for any node remains the same in all networks.

Formally, a motif is defined as an induced subgraph g_k of a graph G when for a given set of parameters $\{P, U, D, N\}$ and a random ensemble of N similar networks, following conditions hold:

- **Minimum frequency** ensures if a subgraph is enough frequent to be considered as a motif. The frequency of the motif on the original network should be higher than an uniqueness threshold U. In a mathematical form, it is as follows:

$$f_{original}(g_k) \geq U \tag{7.1}$$

- **Minimum deviation** is that the frequency of the motif on the original network is significantly larger than its average frequency on the similar random networks. This prevents the detection of motifs that have a small difference between these two values but have a narrow distribution in the random networks. This condition can be formulated as:

$$f_{original}(g_k) - f_{random}(g_k) > D \times \overline{f}_{random}(g_k) \tag{7.2}$$

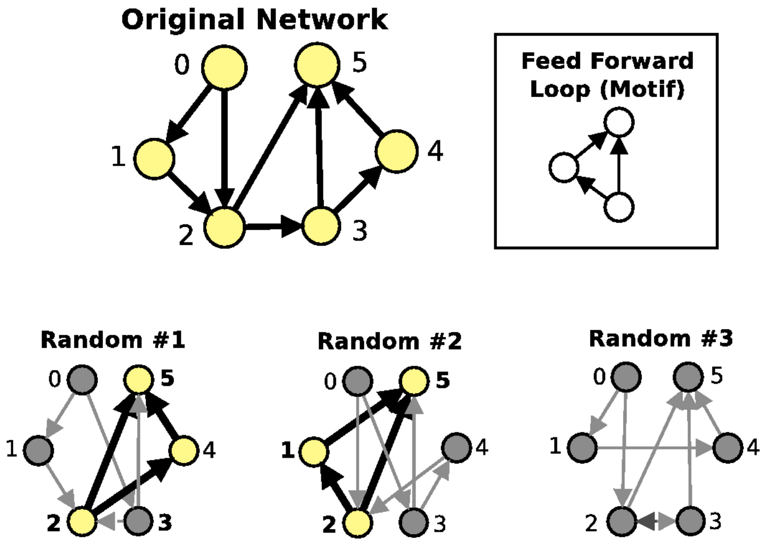


Figure 7.1: An example of network motif of size 3. The random networks preserve the ingoing and outgoing degrees of each node in the original network. The example motif appears at most once in each random network, but has three occurrences on the original one. [RS13]

where D is a proportional deviation threshold that ensures the minimum difference between $f_{original}$ and $\overline{f}_{random}$

- **Over-representation** assures that the occurrence of a motif in a network is not due to global and local properties of the network but it is specific to that particular network. This quantitatively is translated into a statistical hypothesis test where the null hypothesis is: the frequency of a motif in a randomized network is greater than the frequency in the original network for the significance value of P:

$$Prob(\overline{f}_{random}(g_k) > f_{original}(g_k)) \leq P \tag{7.3}$$

The test statistics z-score is defined as equation 7.4 by empirically counting the consensus of the subgraph g_k in an ensemble of a large number of similar random networks and in the original network.

$$z - score(g_k) = \frac{f_{original} - \overline{f}_{random}}{\sigma} \tag{7.4}$$

where $f_{original}$ is the frequency g_k in the original network, $\overline{f}_{random}$ is the average of the frequency g_k in random networks and σ is the standard deviation.

In the seminal paper by Milo et al. [MSOI+02], the parameters $\{P, U, D, N\}$ are set respectively as $\{0.01, 4, 0.1, 1000\}$. This parameter setting can be read as: a subgraph is considered as a motif if: 1) it occurs at least 4 times in the original network; 2) the difference between its frequency in the original network and the average frequency in 1000 random networks is at least 10% of that average frequency; 3) the probability that it appears more often in a random network than in the original network is less than 1%.

7.1.2 Weighted networks

In a weighted network, one requires a significance measure different from the frequency of subgraphs in order to take advantage of available information of weight. Saramaki et al. [OSKK05] used the average of weights to find motifs in a network. They define two measures, *intensity* and *coherence*, based on the average of weights in instances of a particular subgraph.

$$I(g) = (\prod_{(ij)\in l_g} w_{ij})^{1/|l_g|} \tag{7.5}$$

The measure defines a range of subgraph intensities, where zero or very low intensity values imply that the subgraph in question does not exist or exists at a insignificant intensity level. However, intensity cannot quantify the range of weight values in a subgraph. For example, the subgraph intensity $I(g)$ may be low because one of the weights is very low, or it may result from all of the weights being low. In order to distinguish between these two extremes, Saramaki et al. introduced the coherence measurement for a subgraph as:

$$Q(g) = \frac{I(g) * |l_g|}{\sum_{(ij)\in l_g} w_{ij}} \tag{7.6}$$

The coherence value is close to one if the subgraph weights do not differ much, i.e. they are internally coherent.

The total intensity I_M of a motif M in the network is just the sum of its subgraph intensities :

$$I_M = \sum_{g\in M} I(g) \tag{7.7}$$

Following the same notion as for binary motifs, a motif significance is measured by the z-score as follows:

$$z_M = \frac{I_M - \mu(i_M)}{\sigma(i_M)} \tag{7.8}$$

where i_m is the total intensity of motif M in one random network. Analogue to the motif intensity score, they defined the motif coherence score as:

$$z'_M = \frac{Q_M - \mu(q_M)}{\sigma(q_M)} \tag{7.9}$$

where Q_M and q_M are the total coherence for motif M respectively in the original network and in the random networks. A subgraph is a motif if these measurements differ from random values.

7.2 Motif mining methods

7.2.1 Binary networks

Motif mining methods for unweighted networks mainly fall into two main conceptual approaches. Network-centric methods look for all possible k-sized subgraphs, by enumerating connected sets of k vertices, and in the end they do tests to discover the isomorphic class of each subgraph found. ESU [Wer06] and Kavosh [KAE$^+$09] are examples of two state of the art algorithms following this methodology. Subgraph-centric approaches, on the other hand, query individual subgraphs one at the time. Grochow and Kellis [GK07] developed an efficient algorithm for this.

Ribeiro and Silva developed a new specialized data structure, g-tries [RS10], that can efficiently represent and query any collection of subgraphs, following an intermediate set-centric approach, in which we define the custom set of subgraphs we are interested in. G-tries are multiway trees that take advantage of common substructures in the subgraphs to efficiently search at the same time for occurrences of all the subgraphs in the collection. G-tries have been shown to be significantly faster than previous methods when finding motifs [RS10, RS12]. As we will use this method for subgraph enumeration in the following chapter, we study it in more details here.

7.2.1.1 G-tries definition

Ribeiro and Silva [RS13] defined g-trie as a multiway tree that can store a collection of graphs. Each tree node contains information about a single graph vertex, its corresponding edges to ancestor nodes and a boolean flag indicating if that node is the last vertex of a graph. A path from the root to any g-trie node corresponds to one single distinct graph. Descendants of a g-trie node share a common subgraph. In a g-trie all graphs with common ancestor tree nodes share common substructures that are characterized precisely by those ancestor nodes. A single path through the tree corresponds to a different single graph. Children of a node correspond to the different graph topologies that can emerge from the same subgraph. Graphs of different sizes can be stored in the same tree if each tree node

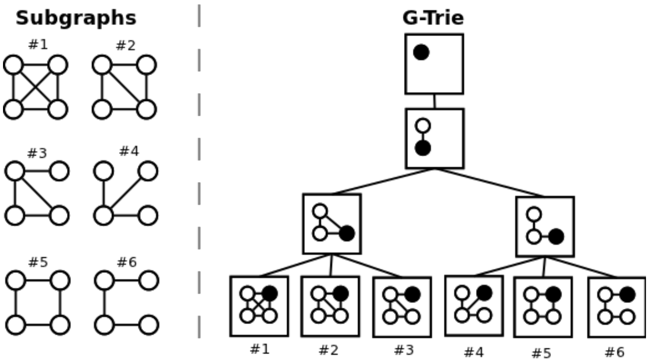


Figure 7.2: An example g-trie storing all possible undirected subgraphs of size 6. In each g-trie node, the black vertex is the new one being added, and the white vertices are the ones "inherited" from the parent g-trie nodes. [CRS12b]

also signals if it corresponds to the "end" of a graph. All of this is easily applicable to both undirected and directed subgraphs.

7.2.1.2 Building a G-trie

To build a g-trie, a possible option is to iteratively insert one subgraph at a time, starting with an empty tree (just a root node). In each insertion, the tree is traversed to verify if any of the children has the same connections to previous nodes as the graph is being inserted. With each increase in depth, the index of the under process vertex is increased. This process is demonstrated in Figure 7.2.

The insertion is completely defined by the adjacency matrix of the inserted graph. However, it is possible to represent adjacency matrices in many different ways, having the same class of isomorphic graphs. Ribeiro and Silva [RS13] address this issue by using a canonical representation for adjacency matrices. They proposed an efficient canonical representation, GTCanon, based on the `nauty` tool [M$^+$81], but adapted to produce the most compact g-trie possible, identifying as much common substructure as possible.

7.2.1.3 Subgraph enumeration using G-trie

Once a g-trie is built, it can be used to find instances of its stored graphs as subgraphs of the original network. By combining an efficient canonical labeling procedure and symmetry breaking conditions, it allows the search at the same time for an entire set of subgraphs. This avoids the redundancy of searching several times for the same substructure that belongs to different subgraphs, as it would happen if we would search for each subgraph type individually, in a subgraph-centric algorithm such as Grochow and Kellis [GK07]. At the same time, g-tries also do isomorphism testing as we are traversing the g-trie tree, since when we are a at a leaf we can be certain that the subgraph found is of that type. This contrasts with network-centric methods such as ESU [Wer06], which enumerate all connected sets of the desired number of vertices and postpone isomorphism tests to when an entire occurrence is found, not reusing information from previous isomorphisms found.

7.2.1.4 Motif discovery

The exact network motifs algorithms, generally calculate a census of subgraphs of a determined size k in the original network. Then, in order to assess the significance of the subgraphs present in the original network a set of similar random networks is generated and the same census is calculated on all of the random networks. Finally the significance score is calculated regarding the equation 7.4. The random networks are normally generated by a Markov chain process [MSOI$^+$02], the execution time of this step is just a very small fraction of the time that the census takes. Computing the census on all random networks is the main bottleneck of the whole process (there can be hundreds of random networks) and g-tries helps precisely in this phase.

7.2.2 Weighted networks

No specific algorithm is proposed for motif mining in weighted networks. Saramaki et al. [OSKK05] used a triangle counting algorithm to find subgraphs of three nodes. They did not consider larger subgraph for weighted motifs. In the next chapter we show how the g-tries algorithm can be adapted for weighted networks. In addition, we propose a new analytical method to find subgraphs census on random networks which significantly decreases the computation time. As discussed in previous section this step is a bottleneck of motif mining algorithms.

Chapter 8

Motif mining in weighted networks

How can motif mining be extended for weighted networks? How do network comparison and classification benefit from motif mining?

For a better characterization of complex networks, one needs to utilize all available information, including the weights of the edges. This is important in networks such as the traffic flow in a transportation network, strength of social relations, or connectivity strength between every pair of brain regions. To find patterns in weighted networks, the majority of the existing methods need a weight threshold over edges to convert a weighted network to an unweighted one, where nodes are connected if the weight is more than the threshold. A big challenge for this approach is to find an appropriate value for the threshold, and different choices of values lead to very different network topologies. For example, two nodes that are connected in a network for threshold a, might be disconnected in a network with threshold b. A limited number of methods were designed to find motif mining considering the weight information and solve this issue [JCZ10, EBH08], as reviewed in chapter 7. The proposes methods mainly use a weighted support measure for frequent subgraph mining algorithms, based on average weights.

In a weighted network, one requires a measure different of the usual frequency to assess the importance of the subgraphs regarding the weight. In motif mining for weighted networks analogous to binary definition, we consider a subgraph prominent if the value of the designed measure is significantly different from its expected value in random network.

In this chapter, we propose two new significance scores to find motifs in weighted networks regarding the weight distribution. We define of a motif as a subgraph that contains unexpected information from a random network, and we define new measurements to assess the exceptionality of subgraphs. We use the g-trie data structure [RS13] to find instances of k-size subgraphs and to calculate their significance score. Following statistical approaches, we find the random score of subgraphs, avoiding the time consuming step of random network generation.

Motif discovery using any algorithm is beneficial to many network study tasks such as to compare networks and to predict the type of the network. We can build motif profiles that can be used as fingerprints for network comparison and classification in different domains such as biological [MIK$^+$04] or social [CRBS12] networks. We show that incorporating the weight information in motif mining algorithms can find the right subgraphs as motifs that best represent the functionality or class of the network. We design two different experiments: network comparison and network classification; as evaluation methods for our proposed significance measurements.

In our first evaluation experiment, we study *Gene co-expression networks* (GCNs) which represent the relationships between genes. We show that how motif profiles can compare GCNs across healthy and cancer-related tissues (section 8.3.1). In the second evaluation experiment, we study co-authorship networks in the biology and mathematics fields. We design a classification problem to predict the class of ego networks of co-authorship networks with pre-defined classes and the motif profile is used as a feature vector for classification (section 8.3.2).

8.1 Significance definition in weighted networks

Since edge weights may be continuous values, it is not straightforward to include them in the mining methodology. For a weighted network, we need a measure that not only considers the frequency, but also includes the weight distribution over the edges in a subgraph. In other words, we need a measure that can assess the whole information embedded in a subgraph in order to assign an importance degree for it to be a pattern.

In this section, we introduce two approaches and definitions for subgraph significance in weighted networks. Both proposed methods are based on the weight distribution in subgraphs. The first one, directly uses weight distribution to assess the significance but the second proposed measure uses entropy of weight in subgraphs as a significance measurement. We explain these definitions for weighted motifs in sections 8.1.1 and 8.1.2, respectively.

8.1.1 Distribution based significance measure

Following the definition of motifs in unweighted networks, we define a subgraph as motif if the weights of the edges in the subgraph follow significantly different distribution than a “similar” random distribution. In classical unweighted network motifs, the original null model involved the creation of random networks with the same degree sequence as the original network [MSOI$^+$02]. This is to guarantee that the motif is really a characteristic of the network and not just a consequence of its global topological properties. In a similar way, in our weighted case we also want to maintain certain global characteristic of the individual network we are analyzing, and we use the weight distribution over the whole network as a suitable random model.

Denoting the probability distribution of weights in a network by $P(w)$, the random distribution of weights in a subgraph with h edges is derived from $P_w(sg) = \prod^h P(w)$. Hence, a motif is a subgraph whose actual weight distribution in subgraph sg is different from the random distribution, which uses the weights over the entire network.

There are several methods that can be used for comparing distributions of weights. Two notable examples are and Kulbeck-Leibler distance [Kul68] or the Kolmogorov-Smirnov test [MJ51]. We follow the univariate comparison where weight distributions are compared edge-wise. We use the two sample Kolmogorov-Smirnov test which compares two samples regarding the location and shape of the empirical cumulative distribution functions of the two samples. For the univariate comparison, the actual weight distribution of every different edge type of subgraph sg is compared with the random distribution. The weighted motifs are those subgraphs for which the probability of holding a weight distribution different from the random distribution is higher than a significance value α. Hence a subgraph sg is a motif if:

$$max\{P(F_{empirical}(w_i) = F_{random}(w_i))|i \in E(sg)\} < \alpha \tag{8.1}$$

where $F_{empirical}(w_i)$ and $F_{random}(w_i)$ are respectively the empirical and random distribution function of w_i, weights on edge i and $E(sg)$ is the set of edges in sg, that is, the set of classes of equivalence over all the edges of sg, as is defined in the section 8.2.1.

Note that in this definition only subgraphs having different distributions over all edges are considered as motifs. An alternative would be to define motifs as subgraphs that have at least one edge with a different distribution. In either definition of motifs in weighted networks, the quality of relations within the subgraph is of interest to us, not its quantity in the network. This suits well for applications where the strength of connections is important such as weighted gene co-expression networks (WGCN). We show how this method can help distinguishing healthy networks from cancer-related networks in WGCN in chapter 8.3.1

We define the weighted score of subgraphs as follows:

$$w\text{-}score_k = argmax\{P(KS(w_i))|i \in E(sg)\} \tag{8.2}$$

where $KS(w_i)$ is the Kolmogorov-Smirnov statistic for distribution comparison of weights on edge i and it is equal to the maximum absolute difference between the empirical weight distribution and random distribution:

$$KS(w_i) = \max_{w \in w_i} |F_{empirical}(w) - F_{random}(w)| \tag{8.3}$$

and $P(KS(w_i))$ are the critical values, regarding the distribution of the KS statistic when $F_{empirical}(w_i) = F_{random}(w_i)$. A *weighted motif profile* of the network can then be constructed as a feature vector containing the w-scores of all subgraph types, explained in section 8.2.1. In this approach, significances of subgraphs are calculated without need for random network generation, as in Equation 8.3 $F_{random}(w_i)$ is derived from weight distribution in the whole of original network.

8.1.2 Entropy based significance measure

The second measurement, we propose for assessing significance of subgraphs in based on entropy concept. We use Shannon's concept of information entropy [Sha01] as the significance measure. Information entropy gives a quantitative measure to assess the amount of latent information in different objects. Entropy measures the uncertainty of a variable; the more randomness it has, the higher the entropy is. Entropy is also used as a measure to differentiate random occurrences or noise in datasets. Given this, it fits well in the problem of motif mining where motifs are the ones which appear in different frequencies than it would be expected in randomized networks. An entropy based approach was also successfully used to discover colored motifs in biological networks [AQRH11]. Our approach is however conceptually different, because we incorporate weight information, while this other approach considers unweighted edges and different node classes.

A subgraph is relevant, and characteristic of the network, if its weight entropy differs significantly from the weight entropy in random networks. To calculate the random entropy, we exploit an analytical approach. In this way, we greatly decrease needed computation time, avoiding the costly step of having to do an exhaustive random network generation for assessing subgraph significance.

Information theory assesses how surprising, or unexpected, an observation or an event is. If an event always happens, there is no information gain in detecting this event. Entropy is a function of the probability distribution $P = (p_1, .., p_n)$ where p_i is the probability of occurrence of an event. Defining the occurrence of a subgraph with an edge weight distribution as an event, we can use entropy as a measure to quantify the importance of subgraph for being a pattern. This measure not only considers the weight distribution in the form of probability function, but also assesses the information content of a subgraph.

If X is the random variable describing a particular subgraph g_h^k with k nodes and h edges in a network then it can have different states regarding different edge weight set $\vec{W} = \{w_i \mid i = 1, .., h\}$. The weight entropy of a subgraph is:

$$H_{\vec{W}}(X) = -\int p(X) log(p(X)) \tag{8.4}$$

where $p(X)$ is the probability of occurrence of weight set $\vec{W}$ in the subgraph g_h^k and is given by:

$$p(X) = P(\vec{W} \leq W) \tag{8.5}$$

where W is a vector of upper bounds for weights of edges in the subgraph.

For each particular type of subgraph of size k in a network, we assign a weight entropy that reflects the weight distribution in the subgraph and shows if the distribution is random or describes a property in the network.

In this method, we define a weighted motif as a subgraph whose weight entropy is significantly different from random weight entropy:

$$|H_R - H_{\vec{W}}| > \delta \tag{8.6}$$

where H_R is the weight entropy in random networks, called random entropy and δ is a user-defined threshold to find motifs.

An essential step of unweighted motif mining methods is the random simulation for calculating the mean and variance of a subgraph frequency in similar random networks [RSK09], typically conserving the degree sequence of the original networks. This step is computationally very expensive. In this thesis, for calculating the random entropy, we do not need this exhaustive generation of random networks, but instead we use analytical formulas to find the probability of occurrence of a subgraph g^k with weight set $\vec{W}$ in an Erdös-Rényi (ER) random graph model. This probability is the main element for calculating the random entropy regarding the Equation 8.4 and is equal to:

$$P_{\vec{W}}^{g_h^k} = p(\vec{W}) * \mu(g^k) \tag{8.7}$$

where $p(\vec{W})$ is the probability that edges in g_h^k have weight set $\vec{W} = \{w_i \mid i = 1,..,h\}$ and $\mu(g^k)$ is the probability occurrence of a subgraph g_h^k. The first component, denoted by $p(\vec{W})$ follows the weight distribution in the original network. The joint probability is as follows where the weight of edges in a random network are independent:

$$p(\vec{W}) = P(\{w_i \mid i = 1,..,h\}) = \prod_i^h p(w_i) \tag{8.8}$$

In a random graph G over a set of nodes V, connectivity between every two nodes i and j is independent and identically distributed in the networks. Edges are described by a set of variables $Y = \{Y_{i,j}\}$ for all $i, j \in V$ where $Y_{i,j}$ is 1 if two nodes are connected, and it is 0 if not. This stationary property of process of random network generation entails that the edge distribution in a network is independent of permutation of nodes, meaning the probability of occurrence of an edge between two nodes i and j does not depend on (i, j) (exchangeable assumption). Picard et al. proposed an analytical method to find the probability of occurrence of a motif in every random network where random variable X is iid [PDK$^+$08]. The probability of motif occurrence is independent of the occurrence position. For the ER model, where the exchangeable assumption holds, the probability of occurrence of g^k is as follows:

$$\mu(g^k) = \prod Pr\{(Y_{i,j} = 1)\}^{e_{ij}} = \alpha^h \tag{8.9}$$

where h is the number of edges in g^k and e_{ij} is 1 if nodes i and j are connected and 0 otherwise, for all $i, j \in V(g^k)$. Finally, by substituting the random probability of occurrence subgraph g^k with weight set $\vec{W}$ in formula 8.4, the random weight entropy is equal to:

$$H^R = -\int_{w_i} \alpha^h * f(w_i)^h log((\alpha * f(w_i))^h) \tag{8.10}$$

8.2 Subgraph enumeration

In this section, we describe how the proposed significance measures are incorporated to motif mining process. To implement the weighted methods described in sections 8.1.2 and 8.1.1, we modified the g-tries search algorithm (explained in chapter 7) to find such subgraphs and calculate the weight distribution in subgraphs and measure the significance score. But before describing the graph enumeration process, we first define the type of subgraphs that will be considered as candidates for enumeration.

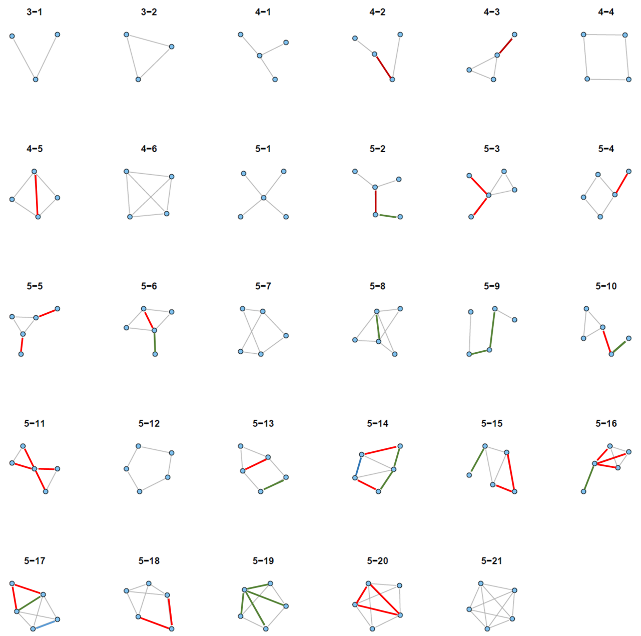


Figure 8.1: Set of subgraphs used for creating a motif profile of the network. Each motif is given an identification that we will use throughout this chapter. Different topological classes of equivalence in the edges of a subgraph are distinguished by color and thickness.

8.2.1 Subgraphs types

In this thesis, we will consider all possible 29 types of undirected subgraphs from sizes 3 to 5 as motif candidates, depicted in Figure 8.1. There is nothing intrinsic in our methodology that forbid us from using even larger sizes, with the exception of potentially being computationally expensive to enumerate all their occurrences.

In each subgraph type we divide its edges in classes of equivalence according to the subgraph symmetry. For instance, there is only one type of edge on the clique of 4 nodes (`4-6` type) since all edges are topologically equivalent. The same can be said for the star subgraph of 4 nodes (`4-1` type). However, in the linear chain of 4 nodes (`4-2` type) there are two different edge types: the one between the middle nodes and the one between a middle node and a leaf node.

8.2.2 Subgraphs enumeration in original network

The overall process for finding motifs of size k in a weighted network is that we first need to find all subgraphs of size k (storing the weight set over the edges for each subgraph type i), and secondly we find the weight distribution over occurrences of subgraph g_i^k with weight set $\{w_1, w_2, ...w_h\}$. This distribution is a multivariate function whose dimension increases as the number of edges in subgraph increases. To find the weight distribution of a given subgraph, we use the stored weight sets while enumerating the instances of the subgraph in the original network, more detailed in section 8.2.3.

We use g-tries [RS10] for storing and searching for subgraph occurrences. G-tries are multiway trees that are able to store a collection of subgraphs. Their basic principle is to identify common substructure. Subgraphs with the same parent g-trie node share the same topological structure with the exception of a single node and its connections.

By using an efficient canonical labeling procedure and symmetry breaking conditions, g-tries allow the search at the same time for an entire set of subgraphs. This avoids the redundancy of searching several times for the same substructure that belongs to different subgraphs, as it would happen if we would search for each subgraph type individually, in a subgraph-centric algorithm such as Grochow and Kellis [GK07]. At the same time, g-tries also do isomorphism testing as we are traversing the g-trie tree, since when we are a at a leaf we can be certain that the subgraph found is of that type. This contrasts with network-centric methods such as ESU [Wer06], which enumerate all connected sets of the desired number of vertices and postpone isomorphism tests to when an entire occurrence is found, not reusing information from previous isomorphisms found.

We modified the original g-tries algorithm so that we are able to store sets of edge weights for each subgraph type, instead of simple integer frequency. After discovering all occurrences of a subgraph g^k in the network, we find its multi-dimensional distribution of weights and calculate significance score of subgraph g^k in the network, regarding measures defined in sections 8.1.2 and 8.1.1.

8.2.3 Empirical weight distribution of subgraphs

An approach to find a distribution is to build the histogram of the data. We use a discretization method to find the histogram, and there are several methods for this purpose. Some of them are supervised methods that need a class label, such as an entropy based method, and others are unsupervised, such as equal width or equal frequency. Here, we use an equal frequency method since we do not have any class label and also because this method finds the intervals that have enough instances for inference, avoiding the generation of sparse intervals in terms of frequency. Equal frequency discretization divides the range of weights for an edge into r intervals where each interval includes n/r values, and

n is the number of weight sets. In this way, we have a set of break points $b_1, ..., b_{r-1}$ and a set of frequency counts that define r intervals in the range of each edge weight: $(-\infty, b_1], [b_1, b_2], ..., [b_{r-2}, b_{r-1}], [b_{r-1}, \infty)$. Label b_i is assigned to values belonging to interval i.

8.3 Evaluation methods

We design experiments to demonstrate the capability of our methods in characterizing networks and to show that incorporating the weight information in motif mining algorithms can find the right subgraphs as motifs that best represent the functionality or class of the network.

We design two set of experiments to evaluate our methods: 1) network comparison and 2) network classification. In the first one we use the weight distribution metric, defined in section 8.1.1 for motif mining in gene co-expression networks and in the second experiment, we employ the entropy-based metric, defined in section 8.1.2 for motif mining in co-authorship networks.

8.3.1 Networks comparison using motif profiles

We evaluate the efficacy of the proposed method for weighted motifs in section 8.1.1 by an comparison experiment. In this section, we study *Weighted Gene Co-Expression Networks* (WGCNs) across healthy tissues and disease associated ones. One important goal of studying WGCNs is to predict gene functions and disease biomarkers such as the discovery of cancer related genes [PHS$^+$07, ZHXJ09]. We particularity seek the differentiating substructure from a healthy network to a cancer related one by comparing networks using *network motifs* as mall connected subgraphs representing characteristic patterns of a network, we use the method explained in section 8.1.1 for network comparison. Our goal is to find *weighted motifs* as sets of differently connected genes in weighted co-expression networks and to use their relative importance as a fingerprint of the network. Our concept of weighted motifs is therefore well suited to applications where the strength of relations between entities is more important, as is the case in WGCNs.

8.3.1.1 Data

The NCBI Gene Expression Omnibus (GEO) is a very rich source for cancer microarray datasets. We queried GEO to retrieve data of various types of tumor biopsy samples. We selected microarray data for three cancer types, including lung cancer, breast cancer and neuroblastoma cancer, as depicted in Table 8.1. All the datasets include at least 30 samples in order to have reliable correlations between genes as mentioned in [OHG06, MDA$^+$09]. We also retrieve two datasets of a normal "healthy" tissue microarray.

Table 8.1: The microarray datasets used for gene co-expression network construction.

GSE NO.	CancerType	SampleSize
GSE12460	neuroblastoma	64
GSE2570	neuroblastoma	38
GSE18864	breast cancer all types	84
GSE21653	medullary breast cancers	266
GSE10445	lung adenocarcinoma	72
GSE3141	lung	111
GSE10245	lung	58
GSE19804	lung	120
GSE10072	lung	107
GSE5056	lung	44
GSE1643	normal	40
GSE13564	normal	44

In the classic unweighted scenario, the co-expression network is constructed with nodes representing genes, and two nodes are connected if the corresponding genes are significantly co-expressed across chosen tissue samples. However, in such network construction it is important to know at what level of correlation two nodes must be connected to be biologically meaningful. Instead of a binary definition of connections between genes (connected=1, unconnected=0), we use a "soft thresholding" framework, as proposed by Zhang and Horvat et al. [ZH05], to build weighted gene co-expression networks, where associated connections have a strength value.

The similarity of genes is measured regarding their gene expression profiles and is used as the weight of connections in the network. Given two genes i and j, the similarity between them, s_{ij}, is defined as the absolute value of the Pearson correlation $s_{ij} = |cor(i,j)|$. Then, the similarity matrix by $S = [s_{ij}]$ is transformed to an adjacency matrix using a thresholding function defined as:

$$a_{ij} = |s_{ij}|^{\beta}$$

where a_{ij} is the weight of the connection between nodes i and j and β is the parameter chosen with the scale-free topology criterion. This is based on the fact that metabolic networks in all organisms have been suggested to be scale-free networks [GIZ$^+$06, CZF$^+$06, DH07].

For each of the microarray dataset in Table 8.1, we build the adjacency matrix of all genes and then extract the network of 500 most connected genes in each dataset. We limit our study to this number of genes as our main concern in this thesis is showing the applicability of our method and not computational issues. The larger the network, the longer the motif mining process will be.

8.3.1.2 Results

We enumerated all 29 subgraphs types, stored the respective set of weights and we proceeded by computing the weighted score of each subgraph using weight distribution method, described in section 8.1.1. Finally, we aggregated all the scores in one feature vector per network, creating an individual fingerprint for each co-expression network.

Figure 8.2 shows the average motif profiles we found on each type of network.

Figure 8.3 is a heat map showing the similarity of gene co-expression networks for healthy tissues and cancer associated networks. The similarity of two networks is measured in terms of Euclidean distance of their weighted motif profiles. From this figure we can clearly see that the weighted motif profile is capable of distinguishing between different network classes. Each network type, including breast cancer, lung cancer, neuroblastoma and healthy tissues, are clearly separated into different groups.

Figure 8.4 shows the most outstanding subgraphs (or motifs) in terms of differentiating gene co-expression networks. These subgraphs are those that make the most difference

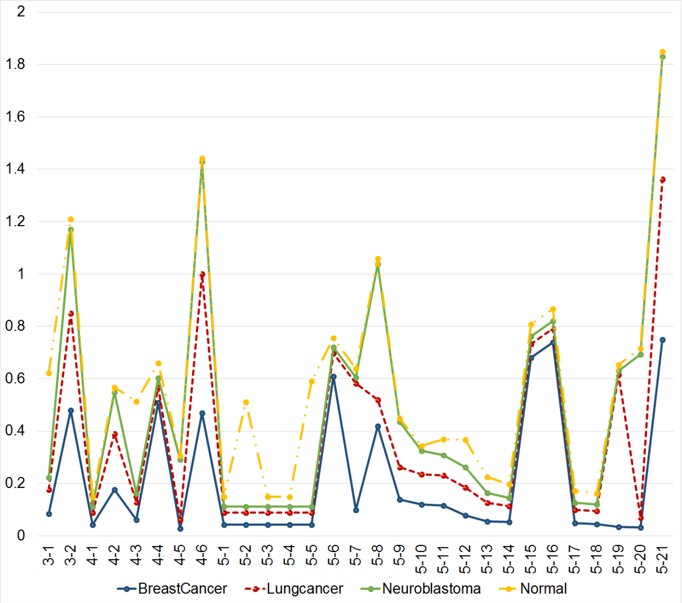


Figure 8.2: Weighted motif profiles of gene co-expression networks for each network type. The subgraphs score is the average for each network type .

between network types regarding motif profiles in Figure 8.2. Less dense subgraphs (`4-3`) and (`5-2`) are more significant in normal networks than the other types. Although in all cancer associated networks dense subgraphs like (`5-21`) are significant, there are some other types of subgraphs that distinguishes them between themselves. Subgraph (`5-7`) for breast cancer, subgraphs (`5-15`) and (`5-16`) for neuroblastoma and subgraph (tt 3-2) for lung cancer are outstanding.

In the next section, we study the significance of weighted network motifs in biological terms and compare binary motifs against our weighted motif profile.

8.3.1.3 Domain based evaluation

We use the domain-based metric to evaluate the discovered motif regarding their biological relevance. Every gene product is described in terms of their association to biological processes, cellular components and molecular functions. A biological process refers to entities at both the cellular and organism levels of granularity, cellular component refers to the localization of proteins inside the cell and molecular function refers to shared activities at the molecular level. The Gene Ontology (GO) [1] database provides vocabularies to describe functions of genes. We use GO term enrichment analysis to find out what function every

[1] http://www.geneontology.org/GO.ontology.structure.shtml

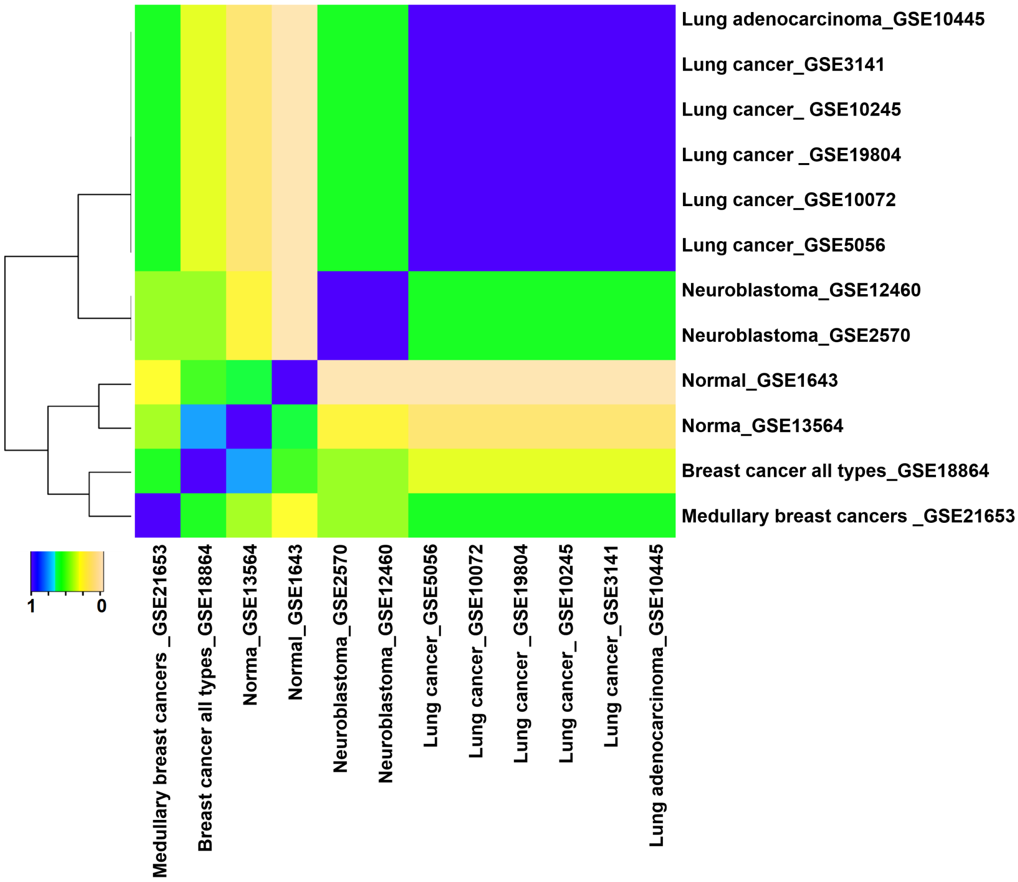


Figure 8.3: Similarity matrix of gene co-expression networks for datasets with 3 types of cancers and 2 healthy cases. The similarity is calculated by Euclidean distance of networks based on their weighted motif profiles.

motif is enriched in.

Only finding the relevant GO terms associated with a given gene list of each motif does not reveal the statistical and biological significance of a function. Hence, we use p-values to assess the chance of observing a particular GO term [AMM05, ABB$^+$00]. If the set of genes participating in motif sg is of size n and m genes have a particular biological annotation then the probability of observing m or more genes, annotated with the same GO term out of n genes is given by:

$$p\text{-}value = \sum_{i=m}^{n} \frac{\binom{M}{i}\binom{N-M}{n-i}}{\binom{N}{n}} \tag{8.11}$$

where N is the number of genes in the database and M is the number of genes that have the same annotation. In other words we are testing the hypothesis of a motif being associated to a particular biological annotation or not. Smaller p-values show that the association is

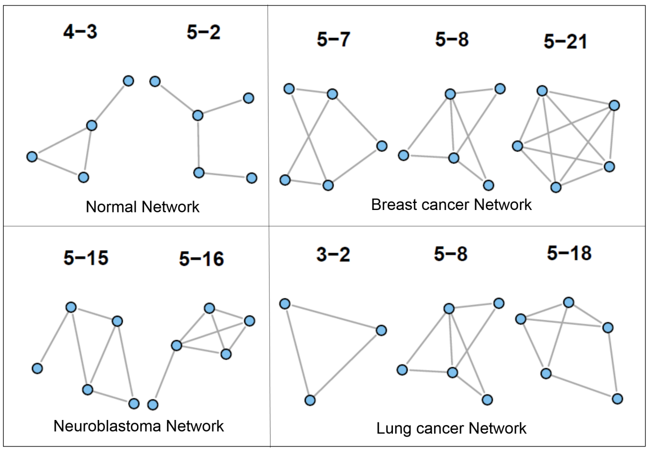


Figure 8.4: Discriminating subgraphs for each type of networks.

not random and is biologically more significant than one with a higher p-value. We can distinguish biologically significant motifs from non-significant ones using a cutoff then we compare different motif profiles (binary and weighted) regarding the scoring function:

$$\text{Motif profile score} = 1 - \frac{\sum_{i=1}^{n_S} \min(p_i) + n_I * cutoff}{(n_S + n_I) * cutoff} \tag{8.12}$$

where n_S and n_I are respectively the number of significant and insignificant motifs and $\min(p_i)$ denotes the smallest p-value of the significant motif i. A motif with a p-value less than a cutoff is significant.

The motif profiles (binary/weighted) are compared using the score function across three ontologies vocabularies namely biological, cellular and molecular. Figure 8.5 shows the comparison between weighted and binary profiles of three cancer types and normal networks. We can see that the weighted profile of a network has higher biological score i.e. the number of motifs discovered by our weighted method are also biologically significant.

8.3.2 Networks classification using motifs profiles

As an evaluation method, we use a classification problem where a set of networks with pre-defined labels or classes are given and the motifs profiles are used as feature vectors for classification. We did the classification with two different scenarios: binary feature vectors and continuous values. In the first scenario, we have a binary vector of size 29 (the number of subgraph types) where we use 1 if the importance value of subgraph is above a defined threshold δ, and we use 0 if it is below the threshold. In the second scenario, we used the original value of motif profiles. For the purpose of comparison we normalize the significance values of both methods into interval of $[-1, 1]$.

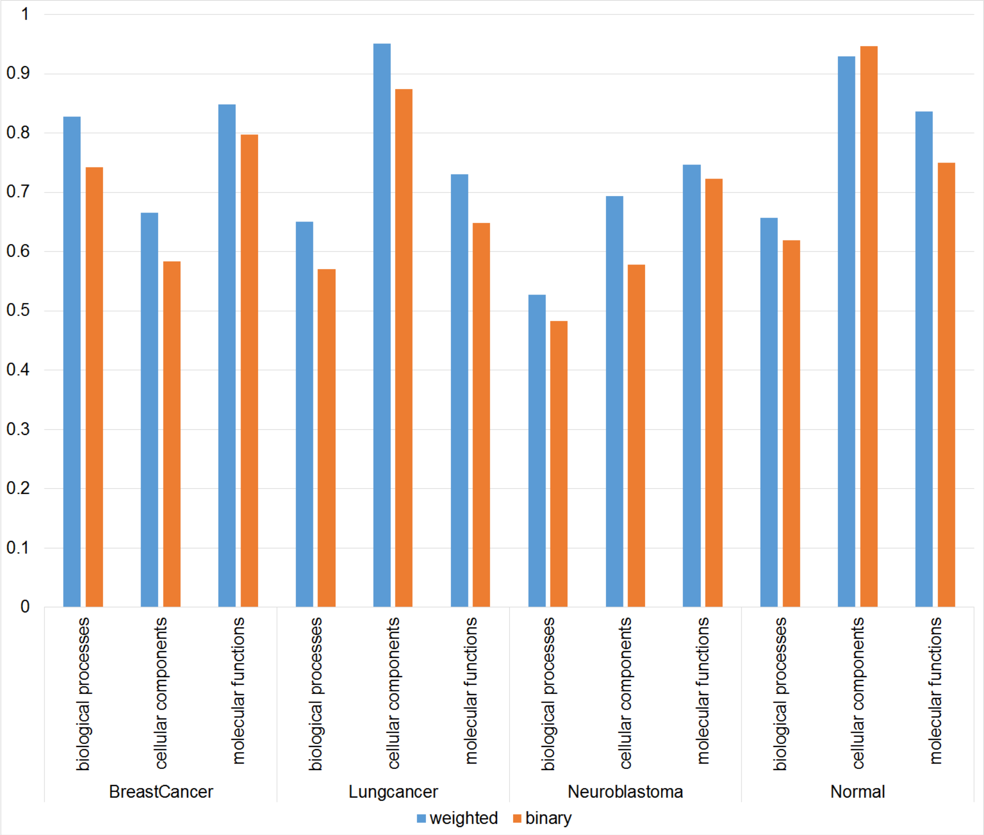


Figure 8.5: Domain base score of motif profiles for three types of cancer and an instance of normal gene co-expression networks .

We apply our proposed method in section 8.1.2 and also the the classical unweighted version of motifs as explained in chapter 7 on the input dataset to derive the motif profiles which are then used as feature vectors for a standard classifier. Then, the accuracy of classification using both the weighted and unweighted methods are compared to assess the obtained performance in finding the correct motifs in the networks.

We use a variety of classification techniques for the evaluation, including: (i) Decision Trees (C4.5) [Qui93], (ii) Naive Bayesian Classifiers (NB) [Mit96], and (iii) Support Vector Machines (SVM) [Vap98]. The classification results were computed using 10-fold cross validation.

The proposed entropy based method in section 8.1.2 and the unweighted one both generate a vector of importance values for subgraphs, respectively called h-score score and z-score. In an unweighted network, the significance of a subgraph is measured in terms of a z-score:

$$z\text{-}score_k = \frac{freq_{original}(G_k) - \overline{freq_{random}(G_k)}}{\sigma(freq_{random}(G_k))}$$

where $\overline{freq_{random}}$ and $\sigma(freq_{random})$ are respectively the average and standard deviation of the frequency in the randomized networks. We derived the motif profile of networks for subgraphs of size 3 to 5 (the usual size in motif mining studies), depicted in Figure 8.1.

8.3.2.1 Data

For our evaluation experiment, we need a dataset of labeled networks. We use the co-authorship networks of publications authored by researchers from the University of Porto, ranging from 2003 to 2011. These are publications drawn from ISI Thompson Web of Knowledge. We randomly selected 100 authors from two different scientific fields: biology and mathematics. Then, for each author, we built the ego net of authors' collaborations, that is, the network composed solely by the authors that have at least one paper co-authored with him, and their respective interconnections (co-authorship of papers). The label of each ego network is the scientific field that the author belongs to. We selected 30% of authors from mathematics and the others from biology. The weight of the edges is the number of papers that two authors published together.

8.3.2.2 Results

Figures 8.6 and 8.7 depict the kernel density estimates of importance scores for the used 100 ego networks in biology and mathematics fields. The plots give the probability that the score of a subgraph fall in an interval. Although both measures give very similar results, h-score values are more concrete and less scattered. As we can see in the figures, h-scores of subgraphs are more centralized around the mean value of importance measure. Hence, if a subgraph is a promising feature in a network, h-score tends to give a stronger value to it. In the figures, the red vertical baselines show the threshold of ± 0.6. Regarding the baselines, we can see that if a subgraph is a motif the h-score can detect it with higher probability than z-score. Comparing the histograms across the two research fields, biology and mathematics, we can see clearly that both measures give higher score to different sets of subgraph for each field. For example, subgraphs of size 5 have higher average importance in biology, specially subgraph 5-20 and 5-21 which are more connected. While in mathematics, the average score for smaller and less connected subgraphs, such as 4-1 and 5-1, is higher. The observed pattern for these two fields are in good accordance with results derived in our previous work [CRBS12] where co-authorship networks are compared across different scientific fields by their motif profile.

The accuracy of built models for two motif mining methods are compared in table 8.2. The

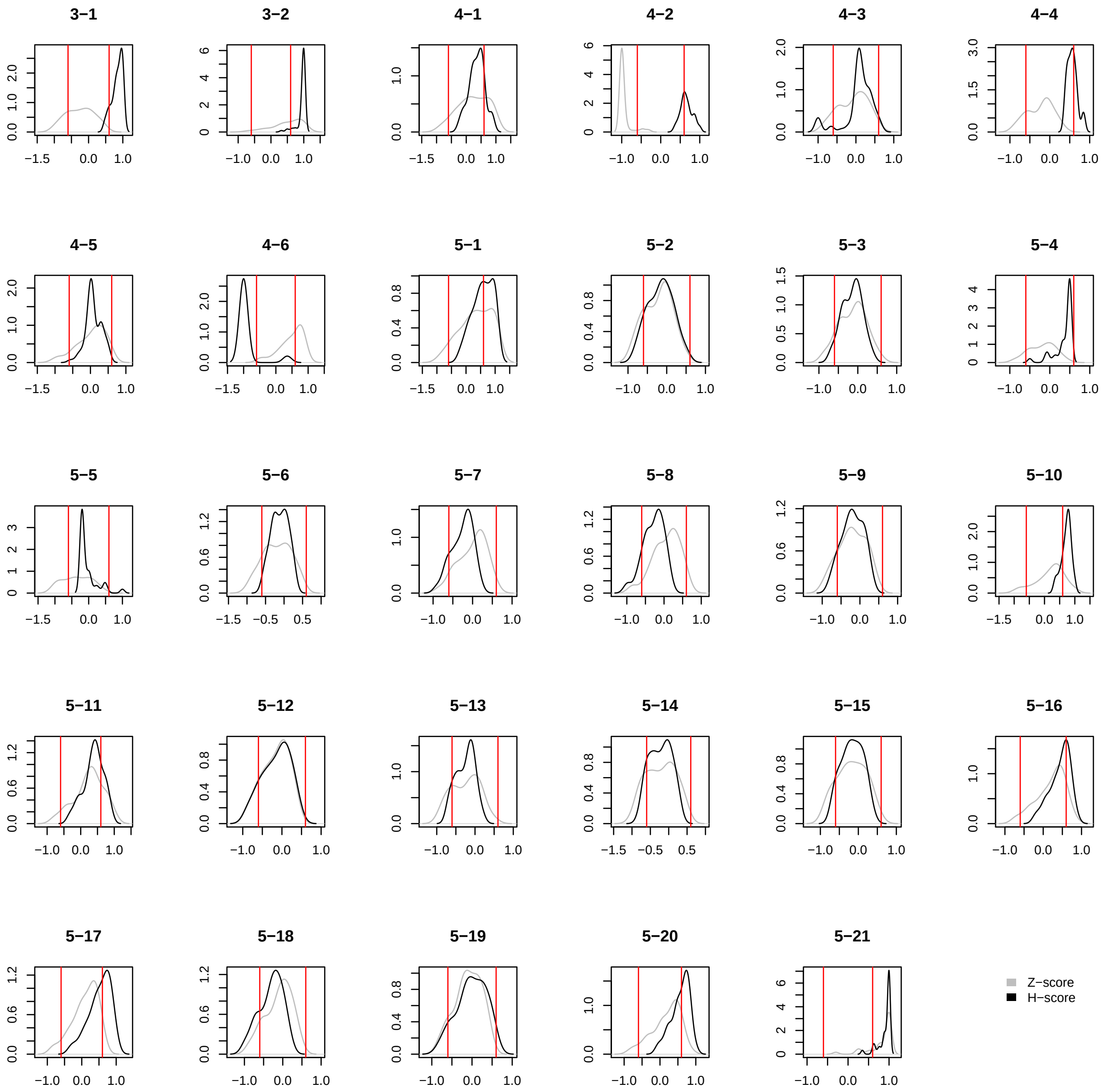


Figure 8.6: Kernel density estimate of significance scores, h-score and z-score, for subgraph size 3-5 for biology ego networks. The red vertical base lines depict the threshold of ±0.6 to consider a subgraph as a motif.

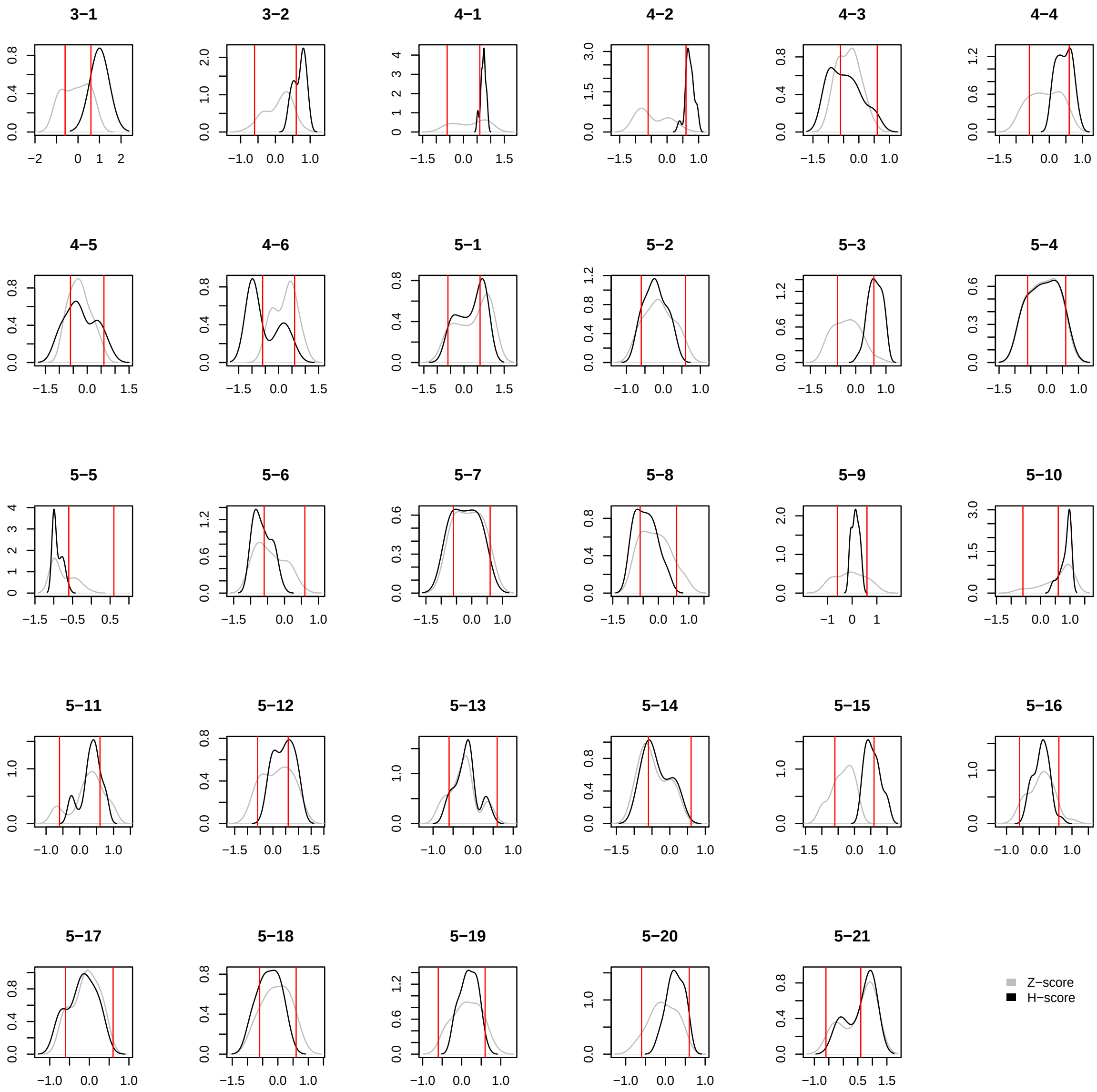


Figure 8.7: Kernel density estimate of significance scores, h-score and z-score, for subgraph size 3-5 for mathematics ego networks. The red vertical base lines depict the threshold of ±0.6 to consider a subgraph as a motif.

last row of the table shows the results for the case in which we used the continuous values of motif profiles.

Table 8.2: The accuracy of the classifiers using the new proposed weighted motif mining method and the classical unweighted method.

threshold δ	weighted motifs			unweighted motifs		
	C4.5	NB	SVM	C4.5	NB	SVM
0.2	80.7	74.1	69.3	71.2	69.4	64.2
0.4	79.9	72.7	72.3	76.5	73.6	67.8
0.6	81.2	75.3	71.3	82.1	75.4	68.1
(continuous)	71.9	68.3	64.3	65.7	67.8	61.7

From table 8.2, we can see that both methods achieve reasonably good results. Compared to the unweighted version of motifs, the proposed method, not only can characterize the networks, but can also do it with slightly better accuracy. In addition, it has two advantages. First, it takes advantage of weight information in the networks and there is no need for putting a threshold over the weight of edges. Secondly, since this method mainly relies on the distribution of weight in the network, we could use statistical methods to calculate the random value of entropy and we avoid the expensive computational step of motif mining algorithm, which is the random network simulation and correspondent subgraph frequency computation, for measuring the z-score.

8.4 Conclusions

Many real complex networks contain more connectivity information than a simple boolean function that tells us whether a pair of nodes is connected or not. The edges can have weights that greatly improve the expressiveness and information content of the connections. For instance, on co-autorship networks, an unweighted network would not distinguish a connection between two authors that wrote dozens of papers together from a connection between a pair of authors that only were co-authors on a single paper. The same concept can be applied in many other network types, expressing for example the amount of traffic flow in a transportation network, or the connectivity strength between brain regions. In this chapter we proposed precisely a novel methodology that is able to find motifs in weighted networks, incorporating the weight information in its calculations.

It is has been shown that subgraph patterns, or motifs, can characterize the functionality of unweighted networks [MIK$^+$04]. We defined motifs in weighted networks as the subgraphs that include unexpected information content, that is, that are different from random networks. We proposed a new significance measure based on weight entropy of subgraphs. In our method, we exploit an analytical approach instead of random networks generation for calculating significance score.

The derived results are compared against unweighted motifs in terms of capability for network characterization. With this purpose in mind, a graph classification problem is used to evaluate the results. The evaluation shows that the proposed method is able to find the set of subgraphs that can differentiate networks at least as well as unweighted motifs, achieving even slightly better accuracy. However, our method is even faster to compute, given that we avoid the random network generation.

This definition is well suited for applications such as gene co-expression networks where the goal is to find groups of genes differentially expressed. In the end we are able to construct a characteristic weighted fingerprint of a network.

We applied our method on several healthy and cancer related datasets to compare the gene networks in terms of their structural patterns, showing that our fingerprint is capable of distinguishing different types of networks. We also showed that the discovered weighted motifs are more biologically relevant when compared to the discovered traditional binary motifs.

Part III

Conclusion and Future directions

Chapter 9

Concluding remarks

"My heart was never deprived of knowledge
few secrets remain that I have not learned,
For seventy-two years I have pondered day and night,
now I know this: Nothing is really known."

– Omar Khayy´am, 1048-1131

9.1 Research Summary and Contributions

The explosive growth in data that we are witnessing naturally opens an enormous opportunity for researchers to develop new methodologies to dynamically extract useful information and knowledge from the data. Real life data inherently contains structural information on objects and their relationships. This structure can be modeled with networks, or graphs, that are abstract representations of a set of nodes and the connections between them.

In this thesis we proposed a series of methodologies to explore connectivity pattern in complex networks, in order to better characterize and understand the networks. We approach this problem from two angles: 1) nodes characterization by *role mining* and 2) subgraphs discovery in weighted networks by *motif mining*.

9.1.1 Role mining

In Part I, we studied the problem of role mining where the goal is to group nodes based on their structural properties in a network. We developed three different methods to study different aspects of role mining namely, role dynamic, evolutionary role mining and relational role mining. We briefly review these methods here:

- Dynamic of roles in a network: We proposed a network characterization method that considers both a static and a dynamic point of view. It is a two phase methodology that automatically assigns labels to nodes of the network based on their local properties and extracts events happening during the evolution of network. The static view provides a general description of the network through label assignment to groups of nodes. Each group in the network is well characterized by the corresponding feature vector profiling. From a dynamic point of view, the methodology discovers rules to describe dynamics of roles, particularly five categories of events are define for each role, emerge, growth, constant, shrink and dissolve. The extracted events are described by some rules that depict the reason of each event and the flow of transition between clusters.
- Relational role mining: We studied patterns of homophily for structural roles in a network. We showed how structural compatibility varies across different structural roles and devise a new method to take advantage of this property for discovering some of structural roles and avoiding misclassification for the others. We proposed a novel relational *structural role mining* method to find roles configuration over a network. Our method is capable of finding roles membership of users regarding their structural features and pairwise dependencies. It iteratively assigns users into structural roles in a way that the derived roles set has the most coherency in terms of including most similar users and has the least non-compatibility of roles in the neighborhood of each user. This algorithm automatically finds the appropriate number of roles in a network by controlling the pairwise dependency parameter.
- Evolutionary role mining: We presented an evolutionary clustering for role extraction in networks. Our method finds the structural role of nodes regarding their current position in the network and their historic data. The role set of nodes at each time step is the one that minimizes the defined cost function for evolutionary clustering, constituting snapshot and historic cost. We utilize the ensemble clustering in our method where nodes at each time step are clustered by aggregating all the available partitionings of data in previous time steps. We use a weighting function to incor-

porate temporal smoothness into the evolutionary clustering method. We conducted an empirical evaluation using normalized mutual information (NMI) and modularity metrics to demonstrate the performance of our method in capturing evolutionary roles in networks. The modularity assess how well roles fit to the current structure of network and NMI metrics evaluate the closeness of current role to previous roles of nodes. The evaluation results on real world networks shows that spectral clustering and hierarchical clustering algorithms outperform HGPA method and have better performance than the baseline approaches as well. In addition, we defined DDW weighting function based on network structure to incorporate temporal aspect of network in role discovery. We showed that this function can better explore evolutionary roles in a network, comparing to temporal weighting function.

- Role mining application in information cascades: There are numerous benefits derived from the notion of role mining, opening up potential application scenarios and research directions. We showed how structural role mining can be applicable to categorize users in information propagation. In this thesis, we also explored the relation between users activity and their structural position. Among structural properties, we found that user commitment in the neighborhood has more impact on the influence score of users in information cascade. In addition, neighborhood cohesion has a more additive effect on users' behavior in terms of blocking a cascade, and triadic closure is also useful. Our experiments showed promising results in terms of correlation between user activities and their temporal structural roles and our model provides a step towards modeling an information cascade independently from the network of diffusion.

9.1.2 Weighted motif mining

In Part II we studied the problem of motif mining in weighted networks. We proposed a new method to incorporate edge weight information in motif mining. We defined a motif as a subgraph that contains unexpected information, and we define new significance measurements to assess this subgraph exceptionality. The proposed metric embeds the weight distribution in subgraphs. We use the g-trie data structure to find instances of k-sized subgraphs and to calculate its significance score. In our method, we exploit an analytical approach instead of random networks generation for calculating significance score.

The discrimination power of the derived motif profile by the proposed method is assessed against the results of the traditional unweighted motifs through a graph classification problem. We use a set of labeled ego networks of co-authorship in the biology and mathematics fields, The new proposed method is shown to be feasible, achieving even slightly better accuracy. Furthermore, we are able to be quicker by not having to generate random networks, and we are able to use the weight information in computing the motif importance, avoiding the need for converting weighted networks into unweighed ones.

We applied our method on several healthy and cancer related datasets to compare the gene networks in terms of their structural patterns, showing that our fingerprint is capable of distinguishing different types of networks. We also showed that the discovered weighted motifs are more biologically relevant when compared to the discovered traditional binary motifs.

9.2 Future research directions

We recognize that the novel approaches described in this dissertation can be developed in a number of ways and open many opportunities for future work. We suggest the following research directions for each part of this thesis.

9.2.1 Role mining

In the section we discuss how role mining methods can be extended in the future research.

- Scaling up role discovery methods: The majority of traditional graph-based role methods were only suitable for relatively small networks. A systematic investigation into these methods and relative parallel speedups would be extremely useful. We note that role features in Algorithm 1 may be computed independently for each node in parallel while role definitions may also be learned in parallel [YHSD12]. Down-sampling (or network sampling) is another promising direction for feature-based role methods [SWM05, ANK13], for instance, feature definitions may be learned on a much smaller sampled network, then roles may be assigned to the sampled nodes based on this feature representation.

- Community based role discovery: One of the emerging challenges in structural role mining is spotting roles of a users relative to the community they belong to. As a future work we intend to extend our method in a way to be capable of finding roles of users in each community they are part of.

- Semi- supervised role mining methods: Another promising direction that has yet to be addressed is following a semi-supervised methodology for role mining. For example, one may compute strict properties of the nodes in the graph and use these to label examples of roles in the graph for which other roles can be learned and extracted. These could now be used in a semi-supervised fashion to help moderate the role discovery algorithm leading to roles that are more interpretable and useful.

9.2.2 Weighted motif mining

Here, we will point out two issues that deserve further investigation in weighted motif mining line of research.

- Random generation of weighted subgraphs: One important step of weighted motif mining is generation of random weighted entropy which has two component: 1) random weight distribution 2) probability of occurrence of a subgraph in a random network. In our proposed method we developed the equation by using Erdös-Rényi model as random graph. This equation might be different for other random networks and furture investigation is required to develop the calculation equation for other models such a scale free.
- Scaling up motif mining methods: The bottleneck of our proposed motif mining models is subgraph enumeration is the original network. This prevent us from experimenting on larger network with millions nodes or using larger subgraph than size 5. This limitation technically is a common issue for all developed subgraph counting methods which is an urging need to be addressed in this research line.

References

[AA04] I. Albert and R. Albert. Conserved network motifs allow protein–protein interaction prediction. *Bioinformatics*, 20(18), 2004.

[AA05] Lada Adamic and Eytan Adar. How to search a social network. *Social Networks*, 27(3), 2005.

[ABB$^+$00] Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1), 2000.

[AIS93] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *ACM SIGMOD Record*, volume 22, 1993.

[AJB99] Réka Albert, Hawoong Jeong, and Albert-László Barabási. Internet: Diameter of the world-wide web. *Nature*, 401(6749), 1999.

[Aka98] Hirotogu Akaike. Information theory and an extension of the maximum likelihood principle. In *Selected Papers of Hirotugu Akaike*. Springer, 1998.

[ALTY08] Nitin Agarwal, Huan Liu, Lei Tang, and Philip S Yu. Identifying the influential bloggers in a community. In *Proc. ACM Int. Conf. on Web Search and Data Mining*, 2008.

[AMM05] Vicente Arnau, Sergio Mars, and Ignacio Marín. Iterative cluster analysis of protein interaction data. *Bioinformatics*, 21(3), 2005.

[ANK13] Nesreen K. Ahmed, Jennifer Neville, and Ramana Kompella. Network sampling: From static to streaming graphs. *ACM Trans. Knowl. Discov. Data*, 8(2), 2013.

[APU07] Sitaram Asur, Srinivasan Parthasarathy, and Duygu Ucar. An event-based framework for characterizing the evolutionary behavior of interaction graphs. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2007.

[AQRH11] Christoph Adami, Jifeng Qian, Matthew Rupp, and Arend Hintze. Information content of colored motifs in complex networks. *Artificial Life*, 17(4), 2011.

[AS94] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. Int. Conf. on Very Large Data Bases*, 1994.

[AS95] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. IEEE Int. Conf. on Data Engineering*, 1995.

[AUP07] Sitaram Asur, Duygu Ucar, and Srinivasan Parthasarathy. An ensemble framework for clustering protein–protein interaction networks. *Bioinformatics*, 23(13), 2007.

[AY05] C.C. Aggarwal and P.S. Yu. Online analysis of community evolution in data streams. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2005.

[BA99] A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439), 1999.

[Bar14] Albert-László Barabási. *Linked: How everything is connected to everything else and what it means for business, science, and everyday life*. Basic Books, 2014.

[BBBG09] Michele Berlingerio, Francesco Bonchi, Björn Bringmann, and Aristides Gionis. Mining graph evolution rules. In *Machine Learning and Knowledge Discovery in Databases*. Springer, 2009.

[BBHM13] Raquel A Baños, Javier Borge-Holthoefer, and Yamir Moreno. The role of hidden influentials in the diffusion of online information cascades. *EPJ Data Science*, 2, 2013.

[BCFM00] Andrei Z Broder, Moses Charikar, Alan M Frieze, and Michael Mitzenmacher. Min-wise independent permutations. *Journal of Computer and System Sciences*, 60(3), 2000.

[BCM11] Smriti Bhagat, Graham Cormode, and S Muthukrishnan. Node classification in social networks. In *Social network data analytics*. Springer, 2011.

[BEF84] James C Bezdek, Robert Ehrlich, and William Full. Fcm: The fuzzy< i> c</i>-means clustering algorithm. *Computers & Geosciences*, 10(2), 1984.

[Ber06] Pavel Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*. Springer, 2006.

[Bes86] Julian Besag. On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1986.

[BGD$^+$06] C. Bird, A. Gourley, P. Devanbu, M. Gertz, and A. Swaminathan. Mining email social networks. In *Proc. ACM Int. workshop on Mining software repositories*, 2006.

[BHKL06] L. Backstrom, D. Huttenlocher, J. Kleinberg, and X. Lan. Group formation in large social networks: membership, growth, and evolution. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2006.

[BHMW11] Eytan Bakshy, Jake M Hofman, Winter A Mason, and Duncan J Watts. Everyone's an influencer: quantifying influence on twitter. In *Proc. ACM Int. Conf. on Web Search and Data Mining*, 2011.

[BKERF12] Michele Berlingerio, Danai Koutra, Tina Eliassi-Rad, and Christos Faloutsos. Netsimile: a scalable approach to size-independent network similarity. *arXiv preprint arXiv:1209.2684*, 2012.

[BL97] Michael J Berry and Gordon Linoff. *Data mining techniques: for marketing, sales, and customer support*. John Wiley & Sons, Inc., 1997.

[BL06] J. Berg and M. Lässig. Cross-species analysis of biological networks by bayesian alignment. *Proceedings of the National Academy of Sciences*, 103(29), 2006.

[BMA83] Ronald S Burt, Michael J Minor, and Richard D Alba. *Applied network analysis: A methodological introduction*. Sage Publications Beverly Hills, CA, 1983.

[BMS97] S. Brin, R. Motwani, and C. Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *ACM SIGMOD Record*, volume 26, 1997.

[Bon07] Phillip Bonacich. Some unique properties of eigenvector centrality. *Social Networks*, 29(4), 2007.

[CBAG12] Meeyoung Cha, Fabrício Benevenuto, Yong-Yeol Ahn, and Krishna P Gummadi. Delayed information cascades in flickr: Measurement, analysis, and modeling. *Computer Networks*, 56(3), 2012.

[Cha07] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2), 2007.

[CHBG10] Meeyoung Cha, Hamed Haddadi, Fabricio Benevenuto, and P Krishna Gummadi. Measuring user influence in twitter: The million follower fallacy. In *Proc. AAAI Int. Conf. on Weblogs and Social Media*, volume 10, 2010.

[CKT06] D. Chakrabarti, R. Kumar, and A. Tomkins. Evolutionary clustering. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Philadelphia, USA, 2006.

[CMG09a] Meeyoung Cha, Alan Mislove, and Krishna P. Gummadi. A Measurement-driven Analysis of Information Propagation in the Flickr Social Network. In *Proc. ACM Int. Conf. on World Wide Web*, 2009.

[CMG09b] Meeyoung Cha, Alan Mislove, and Krishna P Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proc. ACM Int. Conf. on World Wide Web*, 2009.

[CPC08] Andrzej Cichocki, Anh Huy Phan, and Cesar Caiafa. Flexible hals algorithms for sparse non-negative matrix/tensor factorization. In *Proc. IEEE Workshop on Machine Learning for Signal Processing*. IEEE, 2008.

[CRBS12] Sarvenaz Choobdar, Pedro Ribeiro, Sylwia Bulga, and Fernando Silva. Co-authorship network comparison across research fields using motifs. In *Proc. IEEE/ACM Int. Conf. on Advances in Social Networks Analysis and Mining*, 2012.

[CRHK09] L.F. Costa, F.A. Rodrigues, C.C. Hilgetag, and M. Kaiser. Beyond the average: detecting global singular nodes from local features in complex networks. *Europhysics Letters (EPL)*, 87(1), 2009.

[CRS12a] Sarvenaz Choobdar, Pedro Ribeiro, and Fernando Silva. Event detection in evolving networks. In *Proc. IEEE Int. Conf. on Computational Aspects of Social Networks (CASoN)*, São Carlos, Brazil, 2012.

[CRS12b] Sarvenaz Choobdar, Pedro Ribeiro, and Fernando Silva. Motif mining in weighted networks. In *Proc. IEEE Int. Conf. on Data Mining*, 2012.

[CRTB07] L da F Costa, Francisco A Rodrigues, Gonzalo Travieso, and PR Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1), 2007.

[CSR11] S. Choobdar, F. Silva, and P. Ribeiro. Network node label acquisition and tracking. In *Proc. Portuguese Conf. on Artificial Intelligence, Progress in Artificial Intelligence*, 2011.

[CSRPar] Sarvenaz. Choobdar, Fernando. Silva, Pedro. Ribeiro, and Srinivasan. Parthasarathy. Dynamic inference of social roles in information cascades. *Data Mining and Knowledge Discovery*, to appear.

[CSSX09] Graham Cormode, Vladislav Shkapenyuk, Divesh Srivastava, and Bojian Xu. Forward decay: A practical time decay model for streaming systems. In *Proc. IEEE Int. Conf. on Data Engineering*, 2009.

[CSZ$^+$07] Y. Chi, X. Song, D. Zhou, K. Hino, and B.L. Tseng. Evolutionary spectral clustering by incorporating temporal smoothness. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, San Jose, CA, USA, 2007.

[CZF$^+$06] Marc RJ Carlson, Bin Zhang, Zixing Fang, Paul S Mischel, Steve Horvath, and Stanley F Nelson. Gene connectivity, function, and sequence conservation: predictions from modular yeast co-expression networks. *BMC genomics*, 7(1), 2006.

[DBdCD$^+$05] Jesse Davis, Elizabeth Burnside, Inês de Castro Dutra, David Page, and Vítor Santos Costa. An integrated approach to learning bayesian networks of rules. In *Machine Learning: ECML*. Springer, 2005.

[DH07] Jun Dong and Steve Horvath. Understanding network concepts in modules. *BMC Systems Biology*, 1(1), 2007.

[DLR77] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1977.

[DM04] Sergei N Dorogovtsev and José FF Mendes. The shortest path to complex networks. *arXiv preprint cond-mat/0404593*, 2004.

[EBH08] F. Eichinger, K. Böhm, and M. Huber. Mining edge-weighted call graphs to localise software bugs. *Machine Learning and Knowledge Discovery in Databases*, 2008.

[EF05] Elena A Erosheva and Stephen E Fienberg. Bayesian mixed membership models for soft clustering and classification. In *Classification—The Ubiquitous Challenge*. Springer, 2005.

[EK10] David Easley and Jon Kleinberg. Networks, crowds, and markets. *Cambridge Univ Press*, 6(1), 2010.

[EK12] David Easley and Jon Kleinberg. Networks, crowds, and markets: Reasoning about a highly connected world, 2012.

[FFF99] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM Computer Communication Review*, volume 29, 1999.

[FJ02] A.L.N. Fred and A.K. Jain. Data clustering using evidence accumulation. In *Proc. Int. Conf. on Pattern Recognition*, volume 4, Quebec City, Canada, 2002.

[For65] Edward W Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 21, 1965.

[GA05] Roger Guimera and Luis A Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, (7028), 2005.

[GBL10] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. Learning influence probabilities in social networks. In *Proc. ACM Int. Conf. on Web Search and Data Mining*, 2010.

[GERD13] Sean Gilpin, Tina Eliassi-Rad, and Ian N. Davidson. Guided learning for role discovery (glrd): framework, algorithms, and applications. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2013.

[GIZ$^+$06] Peter S Gargalovic, Minori Imura, Bin Zhang, Nima M Gharavi, Michael J Clark, Joanne Pagnon, Wen-Pin Yang, Aiqing He, Amy Truong, Shilpa Patel, et al. Identification of inflammatory gene modules based on variations of human endothelial cell responses to oxidized lipids. *Proceedings of the National Academy of Sciences*, 103(34), 2006.

[GK07] J. Grochow and M. Kellis. Network motif discovery using subgraph enumeration and symmetry-breaking. In *Research in Computational Molecular Biology*. Springer, 2007.

[GL10] Rumi Ghosh and Kristina Lerman. Predicting influential users in online social networks. In *Proc. of KDD workshop on Social Network Analysis (SNA-KDD)*, July 2010.

[GL12] Rumi Ghosh and Kristina Lerman. Rethinking centrality: the role of dynamical processes in social network analysis. *arXiv preprint arXiv:1209.4616*, 2012.

[Gle02] K.S. Gleditsch. Expanded trade and GDP data. *Journal of Conflict Resolution*, 46(5), 2002.

[GMT05] A. Gionis, H. Mannila, and P. Tsaparas. Clustering aggregation. In *Proc. IEEE Int. Conf. on Data Engineering*, 2005.

[GN02] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99(12), 2002.

[GR70] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. *Numerische Mathematik*, 14(5), 1970.

[Gra73] Mark Granovetter. The strength of weak ties. *American journal of sociology*, 78(6), 1973.

[Gra85] Mark Granovetter. Economic action and social structure: the problem of embeddedness. *American journal of sociology*, 1985.

[Grü07] Peter D Grünwald. *The minimum description length principle*. MIT press, 2007.

[GWL11] Stephen Guo, Mengqiu Wang, and Jure Leskovec. The role of social networks in online shopping: information passing, price of trust, and consumer choice. In *Proc. of the 12th ACM conference on Electronic commerce*, 2011.

[HGER$^+$12] K. Henderson, B. Gallagher, T. Eliassi-Rad, H. Tong, S. Basu, L. Akoglu, D. Koutra, C. Faloutsos, L. Li, Y. Matsubara, et al. Rolx: Structural role extraction & mining in large graphs. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Beiging, China, 2012.

[HGL$^+$11] Keith Henderson, Brian Gallagher, Lei Li, Leman Akoglu, Tina Eliassi-Rad, Hanghang Tong, and Christos Faloutsos. It's who you know: graph mining using recursive structural features. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2011.

[HQ79] Edward J Hannan and Barry G Quinn. The determination of the order of an autoregression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 1979.

[HS06] Matthias Heiler and Christoph Schnörr. Controlling sparseness in non-negative tensor factorization. In *Computer Vision–ECCV 2006*. Springer, 2006.

[HW79] JA Hartigan and MA Wong. A k-means clustering algorithm. *Journal of the Royal Statistical Society C*, 28(1), 1979.

[HWP03a] Jun Huan, Wei Wang, and Jan Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. In *Proc. IEEE Int. Conf. on Data Mining*, 2003.

[HWP03b] Jun Huan, Wei Wang, and Jan Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. In *Proc. IEEE Int. Conf. on Data Mining*, 2003.

[IM09] José Luis Iribarren and Esteban Moro. Impact of human activity patterns on the dynamics of information diffusion. *Physical review letters*, 103(3), 2009.

[Ino04] A. Inokuchi. Mining generalized substructures from a set of labeled graphs. In *Proc. IEEE Int. Conf. on Data Mining*, 2004.

[JCZ10] C. Jiang, F. Coenen, and M. Zito. Frequent sub-graph mining on edge weighted graphs. *Data Warehousing and Knowledge Discovery*, 2010.

[Joh67] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3), 1967.

[Jol05] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2005.

[JVB$^+$05] W. Jiang, J. Vaidya, Z. Balaporia, C. Clifton, and B. Banich. Knowledge discovery from transportation network data. In *Proc. IEEE Int. Conf. on Data Engineering*, 2005.

[JW02] R.A. Johnson and D.W. Wichern. *Applied multivariate statistical analysis*, volume 5. Prentice Hall Upper Saddle River, NJ, 2002.

[KAE$^+$09] Z. Kashani, H. Ahrabian, E. Elahi, A. Nowzari-Dalini, E. Ansari, S. Asadi, S. Mohammadi, F. Schreiber, and A. Masoudi-Nejad. Kavosh: a new algorithm for finding network motifs. *BMC bioinformatics*, 10(1), 2009.

[KAKS97] George Karypis, Rajat Aggarwal, Vipin Kumar, and Shashi Shekhar. Multilevel hypergraph partitioning: Application in vlsi domain. In *Proc. of the 34th annual Design Automation Conference*. ACM, 1997.

[KGS04] M. Koyutürk, A. Grama, and W. Szpankowski. An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics*, 20(suppl 1), 2004.

[KK04] M. Kuramochi and G. Karypis. Grew-a scalable frequent subgraph discovery algorithm. In *Proc. IEEE Int. Conf. on Data Mining*, 2004.

[KK05] M. Kuramochi and G. Karypis. Finding frequent patterns in a large sparse graph*. *Data mining and knowledge discovery*, 11(3), 2005.

[KKT03] David Kempe, Jon Kleinberg, and Éva Tardos. Maximizing the spread of influence through a social network. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2003.

[KLPM10] Haewoon Kwak, Changhyun Lee, Hosung Park, and Sue Moon. What is twitter, a social network or a news media? In *Proc. ACM Int. Conf. on World Wide Web*, 2010.

[KMR$^+$94] M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. In *Proc. Int. Conf. on Information and knowledge management*, 1994.

[Kul68] Solomon Kullback. *Information theory and statistics*. Courier Dover Publications, 1968.

[KW06] Gueorgi Kossinets and Duncan J Watts. Empirical analysis of an evolving social network. *Science*, 311(5757), 2006.

[LAH07] Jure Leskovec, Lada A Adamic, and Bernardo A Huberman. The dynamics of viral marketing. *ACM Transactions on the Web (TWEB)*, 1(1), 2007.

[LCZ$^+$08] Yu-Ru Lin, Yun Chi, Shenghuo Zhu, Hari Sundaram, and Belle L Tseng. Facetnet: a framework for analyzing communities and their evolutions in dynamic networks. In *Proc. ACM Int. Conf. on World Wide Web*, 2008.

[LF12] A. Lancichinetti and S. Fortunato. Consensus clustering in complex networks. *Scientific Reports*, 2(336), 2012.

[LG14] Ben London and Lise Getoor. Collective classification of network data. *Data Classification: Algorithms and Applications*, 2014.

[LGS12] Kristina Lerman, Rumi Ghosh, and Tawan Surachawala. Social contagion: An empirical study of information spread on digg and twitter follower graphs. *arXiv preprint arXiv:1202.3162*, 2012.

[LKF05a] J. Leskovec, J. Kleinberg, and C. Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, Chicago, IL, USA, 2005.

[LKF05b] Jure Leskovec, Jon Kleinberg, and Christos Faloutsos. Graphs over time: densification laws, shrinking diameters and possible explanations. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2005.

[LKPM10] Changhyun Lee, Haewoon Kwak, Hosung Park, and Sue Moon. Finding influentials based on the temporal order of information adoption in twitter. In *Proc. ACM Int. Conf. on World Wide Web*, 2010.

[LLM10] Jure Leskovec, Kevin J. Lang, and Michael Mahoney. Empirical comparison of algorithms for network community detection. In *Proc. ACM Int. Conf. on World Wide Web*, 2010.

[LNK07] David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7), 2007.

[M$^+$81] Brendan D McKay et al. *Practical graph isomorphism.* Department of Computer Science, Vanderbilt University, 1981.

[MC12] Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 2(1), 2012.

[MDA$^+$09] Nicole K MacLennan, Jun Dong, Jason E Aten, Steve Horvath, Lola Rahib, Loren Ornelas, Katrina M Dipple, and Edward RB McCabe. Weighted gene co-expression network analysis identifies biomarkers in glycerol kinase deficient mice. *Molecular genetics and metabolism*, 98(1), 2009.

[MGA07] Luke K McDowell, Kalyan Moy Gupta, and David W Aha. Cautious inference in collective classification. In *Proc. AAAI Int. Conf. on Artificial Intelligence*, volume 7, 2007.

[MHC$^+$08] Amy McGovern, Nathan C Hiers, Matthew W Collier, David John Gagne II, and Rodger A Brown. Spatiotemporal relational probability trees: An introduction. In *Proc. IEEE Int. Conf. on Data Mining*, 2008.

[MIK$^+$04] Ron Milo, Shalev Itzkovitz, Nadav Kashtan, Reuven Levitt, Shai Shen-Orr, Inbal Ayzenshtat, Michal Sheffer, and Uri Alon. Superfamilies of evolved and designed networks. *Science*, 303(5663), 2004.

[Mit96] T. Mitchell. Machine learning. In *McGraw Hill*, 1996.

[MJ51] Frank J Massey Jr. The kolmogorov-smirnov test for goodness of fit. *Journal of the American statistical Association*, 46(253), 1951.

[MNHP10] T. Milenković, W.L. Ng, W. Hayes, and N. Pržulj. Optimal network alignment with graphlet degree vectors. *Cancer Informatics*, 9, 2010.

[MP07] S.A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *The Journal of Machine Learning Research*, 8, 2007.

[MSLC01] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, 2001.

[MSOI$^+$02] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594), 2002.

[MZL12] Seth A Myers, Chenguang Zhu, and Jure Leskovec. Information diffusion and external influence in networks. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2012.

[New05] Mark EJ Newman. Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5), 2005.

[NG04] M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2), 2004.

[OHG06] Michael C Oldham, Steve Horvath, and Daniel H Geschwind. Conservation and evolution of gene coexpression networks in human and chimpanzee brains. *Proc. of the National Academy of Sciences*, 103(47), 2006.

[OSKK05] Jukka-Pekka Onnela, Jari Saramäki, János Kertész, and Kimmo Kaski. Intensity and coherence of motifs in weighted complex networks. *Physical Review E*, 71(6), 2005.

[PDK$^+$08] F. Picard, J.J. Daudin, M. Koskas, S. Schbath, and S. Robin. Assessing the exceptionality of network motifs. *Journal of Computational Biology*, 15(1), 2008.

[Pea88] Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[PHS$^+$07] Miguel Angel Pujana, Jing-Dong J Han, Lea M Starita, Kristen N Stevens, Muneesh Tewari, Jin Sook Ahn, Gad Rennert, Víctor Moreno, Tomas Kirchhoff, Bert Gold, et al. Network modeling links breast cancer susceptibility and centrosome dysfunction. *Nature genetics*, 39(11), 2007.

[PS11] Raj Kumar Pan and Jari Saramäki. Path lengths, correlations, and centrality in temporal networks. *Physical Review E*, 84, 2011.

[Qui93] J.R. Quinlan. *C4. 5: programs for machine learning*. Morgan kaufmann, 1993.

[RA14] Ryan A. Rossi and Nesreen K. Ahmed. Role discovery in networks. *IEEE Transactions on Knowledge and Data Engineering*, 26(7), 2014.

[Ras99] Carl Edward Rasmussen. The infinite gaussian mixture model. In *NIPS*, volume 12, 1999.

[RFT13] Ryan Rossi, Sonia Fahmy, and Nilothpal Talukder. A multi-level approach for evaluating internet topology generators. In *IFIP Networking Conference, 2013*. IEEE, 2013.

[RGAH11] Daniel M Romero, Wojciech Galuba, Sitaram Asur, and Bernardo A Huberman. Influence and passivity in social media. In *Proc. of the ECML/PKDD*, 2011.

[RGNH12] R. Rossi, B. Gallagher, J. Neville, and K. Henderson. Role-dynamics: fast mining of large dynamic networks. In *Proc. ACM Int. Conf. on World Wide Web*, Lyon, France, 2012.

[RNGH13] R.A. Rossi, J. Neville, B. Gallagher, and K. Henderson. Modeling dynamic behavior in large evolving graphs. In *Proc. ACM Int. Conf. on Web Search and Data Mining*, 2013.

[RP14] Yiye Ruan and Srinivasan Parthasarathy. Simultaneous detection of communities and roles from large networks. In *Proc. ACM Int. Conf. on Online Social Networks*, 2014.

[RS10] P. Ribeiro and F. Silva. G-tries: an efficient data structure for discovering network motifs. In *Proc. ACM Int. Symposium on Applied Computing*, 2010.

[RS12] P. Ribeiro and F. Silva. Querying subgraph sets with g-tries. In *Proc. ACM SIGMOD Workshop on Databases and Social Networks (DBSocial)*, 2012.

[RS13] Pedro Ribeiro and Fernando Silva. G-tries: a data structure for storing and finding subgraphs. *Data Mining and Knowledge Discovery*, 2013.

[RSK09] P. Ribeiro, F. Silva, and M. Kaiser. Strategies for network motifs discovery. In *Proc. IEEE Int. Conf. on e-Science*, 2009.

[RTU13] Daniel M Romero, Chenhao Tan, and Johan Ugander. On the interplay between social and topical structure. In *ICWSM*, 2013.

[S$^+$78] Gideon Schwarz et al. Estimating the dimension of a model. *The annals of statistics*, 6(2), 1978.

[Sch07] Satu Elisa Schaeffer. Graph clustering. *Computer Science Review*, 1(1), 2007.

[SFPY07] Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2007.

[SG03] A. Strehl and J. Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3, 2003.

[Sha01] C.E. Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(1), 2001.

[SJ03] C.A. Sugar and G.M. James. Finding the number of clusters in a dataset. *Journal of the American Statistical Association*, 98(463), 2003.

[SNK08] Kazumi Saito, Ryohei Nakano, and Masahiro Kimura. Prediction of information diffusion probabilities for independent cascade model. In *Knowledge-Based Intelligent Information and Engineering Systems*. Springer, 2008.

[SOMMA02] S.S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics*, 31(1), 2002.

[SR08] Parag Singla and Matthew Richardson. Yes, there is a correlation:-from social networks to personal behavior on the web. In *Proc. ACM Int. Conf. on World Wide Web*, 2008.

[SWK03] Eran Segal, Haidong Wang, and Daphne Koller. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, 19, 2003.

[SWM05] Michael PH Stumpf, Carsten Wiuf, and Robert M May. Subnets of scale-free networks are not scale-free: sampling properties of networks. *Proc. of the National Academy of Sciences of the United States of America*, 102(12), 2005.

[TAK02] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *18th Conference on Uncertainty in Artificial Intelligence*, 2002.

[TBWK07] Chayant Tantipathananandh, Tanya Berger-Wolf, and David Kempe. A framework for community identification in dynamic social networks. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2007.

[TLJF04] A.P. Topchy, M.H.C. Law, A.K. Jain, and A.L. Fred. Analysis of consensus partition in cluster ensemble. In *Proc. IEEE Int. Conf. on Data Mining*, Brighton, UK, 2004.

[TSWY09] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. Social influence analysis in large-scale networks. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2009.

[Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley, 1998.

[VL07] Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4), 2007.

[VSGL11] Greg Ver Steeg, Rumi Ghosh, and Kristina Lerman. What stops social epidemics? In *Proc. AAAI Int. Conf. on Weblogs and Social Media*, 2011.

[Wer06] S. Wernicke. Efficient detection of network motifs. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 3(4), 2006.

[WFYH03] Haixun Wang, Wei Fan, Philip S Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2003.

[WLJH10] Jianshu Weng, Ee-Peng Lim, Jing Jiang, and Qi He. Twitterrank: finding topic-sensitive influential twitterers. In *Proc. ACM Int. Conf. on Web Search and Data Mining*, 2010.

[WLY14] Pei Wang, Jinhu Lü, and Xinghuo Yu. Identification of important nodes in directed biological networks: A network motif approach. *PloS one*, 9(8), 2014.

[WS98] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684), June 1998.

[WSAL12] Ting Wang, Mudhakar Srivatsa, Dakshi Agrawal, and Ling Liu. Microscopic social influence. In *Proc. SIAM Int. Conf. on Data Mining*, 2012.

[WSP07] C. Wang, V. Satuluri, and S. Parthasarathy. Local probabilistic models for link prediction. In *Proc. IEEE Int. Conf. on Data Mining*, Omaha, NE, USA, 2007.

[WZ13] Yu-Xiong Wang and Yu-Jin Zhang. Nonnegative matrix factorization: A comprehensive review. *Knowledge and Data Engineering, IEEE Transactions on*, 25(6), 2013.

[Yan05] Yuhong Yang. Information theory, inference, and learning algorithms. *Journal of the American Statistical Association*, 100(472), 2005.

[Yao03] YY Yao. Information-theoretic measures for knowledge discovery and data mining. In *Entropy Measures, Maximum Entropy Principle and Emerging Applications*. Springer, 2003.

[YH02a] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *Proc. IEEE Int. Conf. on Data Mining*, 2002.

[YH02b] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *Proc. IEEE Int. Conf. on Data Mining*, 2002.

[YHSD12] Hsiang-Fu Yu, Cho-Jui Hsieh, Si Si, and Inderjit S Dhillon. Scalable coordinate descent approaches to parallel matrix factorization for recommender systems. In *Proc. IEEE Int. Conf. on Data Mining*, 2012.

[YL11] Jaewon Yang and Jure Leskovec. Patterns of temporal variation in online media. In *Proc. ACM Int. Conf. on Web Search and Data Mining*, 2011.

[Zac77] Wayne W Zachary. An information flow model for conflict and fission in small groups. *Journal of anthropological research*, 1977.

[ZBS00] Yongyue Zhang, J Michael Brady, and Stephen Smith. Hidden markov random field model for segmentation of brain mr image. In *Medical Imaging 2000*. International Society for Optics and Photonics, 2000.

[ZH05] Bin Zhang and Steve Horvath. A general framework for weighted gene co-expression network analysis. *Statistical applications in genetics and molecular biology*, 4(1), 2005.

[Zhu05] Xiaojin Zhu. Semi-supervised learning literature survey. Technical report, Carnegie Mellon University, 2005.

[ZHXJ09] Jie Zhang, Kun Huang, Yang Xiang, and Ruoming Jin. Using frequent co-expression network to identify gene clusters for breast cancer prognosis. In *Proc. IEEE Int. Conf. on Bioinformatics, Systems Biology and Intelligent Computing*, 2009.

[ZL13] Yang Zhou and Ling Liu. Social influence based clustering of heterogeneous information networks. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2013.

[ZWY$^+$13] Yuchen Zhao, Guan Wang, Philip S Yu, Shaobo Liu, and Simon Zhang. Inferring social roles and statuses in social networks. In *Proc. ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2013.