GRAPH-BASED LEARNING FOR INFORMATION SYSTEMS

by

Xin Li

A Dissertation Submitted to the Faculty of the

Committee On Business Administration

In Partial Fulfillment of the Requirements
For the Degree of

DOCTOR OF PHILOSOPHY
WITH A MAJOR IN MANAGEMENT

In the Graduate College

THE UNIVERSITY OF ARIZONA

2009

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

THE UNIVERSITY OF ARIZONA
GRADUATE COLLEGE

As members of the Dissertation Committee, we certify that we have read the dissertation

prepared by Xin Li

entitled Graph-Based Learning for Information Systems

and recommend that it be accepted as fulfilling the dissertation requirement for the

Degree of Doctor of Philosophy

____ Date: 04/21/2009
Hsinchun Chen

____ Date: 04/21/2009
Jay F. Nunamaker, Jr.

____ Date: 04/21/2009
Daniel Zeng

Final approval and acceptance of this dissertation is contingent upon the candidate's submission of the final copies of the dissertation to the Graduate College.

I hereby certify that I have read this dissertation prepared under my direction and recommend that it be accepted as fulfilling the dissertation requirement.

____ Date: 04/21/2009
Dissertation Director: Hsinchun Chen

STATEMENT BY AUTHOR

SIGNED: Xin Li

ACKNOWLEDGMENTS

I gratefully thank my advisor, Dr. Hsinchun Chen, for his guidance and encouragement throughout my five years of doctoral study at the University of Arizona. It has been an invaluable opportunity for me to work at the Artificial Intelligence Lab under his direction. I believe that what I have learned from him will significantly benefit my future career. Special thanks to my dissertation committee members, Dr. Jay F. Nunamaker, Jr., Dr. J. Leon Zhao, Dr. Daniel Zeng, and my minor committee member from the Department of Electrical and Computer Engineering, Dr. Salim Hariri, for their guidance and encouragement. I also thank all other faculty members for their support. A special thank you goes to my academic advisor at Tsinghua University, Dr. Cheng Wu, for his advice and encouragement.

My dissertation has been partly supported by the National Institutes of Health/National Library of Medicine (1 R33 LM07299-01) and the National Science Foundation (IIS-0311652; CMMI-0533749; CMMI-0549663; CMMI- 0654232; CMMI-0738803). My work at the Artificial Intelligence Lab has been supported by many colleagues. I thank Dr. Mihail C. Roco of the National Science Foundation, Dr. Jesse Martinez of the Arizona Cancer Center, Dr. David W. Galbraith of the Department of Plant Science, Dr. Zhu Zhang of the Department of Management Information Systems, Dr. Michael Chau at the University of Hong Kong, and Cathy Larson, Chun-Ju Tseng, Dr. Zan Huang, Dr. Hua Su, Dr. Jennifer Xu, Dr. Jiexun Li, Daning Hu, Yiling Lin, Yi-Jen Ho, and Yan Dang from the AI Lab for their collaboration. I also want to thank Ms. Sarah Marshall and Ms. Carol Schumaker for editing my papers.

I would like to thank all my colleagues and friends, including Dr. Byron Marshall, Dr. Daniel McDonald, Dr. G. Alan Wang, Dr. Yilu Zhou, Dr. Jialun Qin, Dr. Sherry Sun, Dr. Yiwen Zhang, Dr. Ming Lin, Dr. Siddharth Kaza, Dr. Rob Schumaker, Wei Chang, Dr. Ahmed Abbasi, Nichalin Suakkaphong, Hsin-min Lu, Tianjun Fu, Runpu Sun, Ping Yan, Manlu Liu, Yulei Zhang, Yida Chen, Sven Thomas, Roopali Wakhare, David Zimbra, Shaokun Fan, Shuo Zeng, Li Fan, and Chun-Neng Huang for their support of my studies.

Most of all, I am grateful for the constant support from my parents Xianfen Meng and Guowang Li, my wife, Fenfang Peng, my sister, Dr. Su Li, and the rest of my family, without whom none of this work would have been possible.

DEDICATION

This dissertation is dedicated to my family.

TABLE OF CONTENTS

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

TABLE OF CONTENTS – *Continued*

LIST OF ILLUSTRATIONS

LIST OF TABLES

ABSTRACT

The advance of information technologies (IT) makes it possible to collect a massive amount of data in business applications and information systems. The increasing data volumes require more effective knowledge discovery techniques to make the best use of the data. This dissertation focuses on knowledge discovery on graph-structured data, i.e., graph-based learning. Graph-structured data refers to data instances with relational information indicating their interactions in this study. Graph-structured data exist in a variety of application areas related to information systems, such as business intelligence, knowledge management, e-commerce, medical informatics, etc. Developing knowledge discovery techniques on graph-structured data is critical to decision making and the reuse of knowledge in business applications.

In this dissertation, I propose a graph-based learning framework and identify four major knowledge discovery tasks using graph-structured data: topology description, node classification, link prediction, and community detection. I present a series of studies to illustrate the knowledge discovery tasks and propose solutions for these example applications. As to the topology description task, in Chapter 2 I examine the global characteristics of relations extracted from documents. Such relations are extracted using different information processing techniques and aggregated to different analytical unit levels. As to the node classification task, Chapter 3 and Chapter 4 study the patent classification problem and the gene function prediction problem, respectively. In Chapter 3, I model knowledge diffusion and evolution with patent citation networks for patent classification. In Chapter 4, I extend the context assumption in previous research and

model context graphs in gene interaction networks for gene function prediction. As to the link prediction task, Chapter 5 presents an example application in recommendation systems. I frame the recommendation problem as link prediction on user-item interaction graphs, and propose capturing graph-related features to tackle this problem. Chapter 6 examines the community detection task in the context of online interactions. In this study, I propose to take advantage of the sentiments (agreements and disagreements) expressed in users' interactions to improve community detection effectiveness. All these examples show that the graph representation allows the graph structure and node/link information to be more effectively utilized in addressing the four knowledge discovery tasks.

In general, the graph-based learning framework contributes to the domain of information systems by categorizing related knowledge discovery tasks, promoting the further use of the graph representation, and suggesting approaches for knowledge discovery on graph-structured data. In practice, the proposed graph-based learning framework can be used to develop a variety of IT artifacts that address critical problems in business applications.

CHAPTER 1. INTRODUCTION

1.1 Knowledge Discovery

Knowledge discovery is the process of extracting implicit, unknown, and potentially useful information from data (Fayyad et al., 1996). In business applications, knowledge discovery approaches are critical to decision making because of the difficulty people have utilizing data directly in a raw form for decision making. In order to be of practical value, data need to be transformed to certain forms (Barlas et al., 2005), limited to specific problems (Carlisle, 2006), analyzed for underlying patterns, and aggregated to knowledge. To handle such a procedure, manual processing would need a great deal of human effort, which is becoming more and more difficult due to the rapidly increasing data volumes.

The advance of information techniques (IT) and the use of information systems have greatly enhanced our ability to collect, digitize, and store data and information in a variety of domains. For example, in business applications, we are able to keep track of most of the daily business activities related to production, transportation, marketing, sales, and accounting operations (Sprott, 2000). In knowledge management, we are able to digitize and store most of the books, patents, papers, and other types of documents (Levy and Marshall, 1995). The advance of Internet/telecommunication techniques has even made it possible to preserve our daily communications conducted online or through telephones (Walther, 1996).

Such massive amounts of data provide us with great potential to extract useful knowledge to address business problems. However, in order to process these large volumes of data, it is necessary to have automatic knowledge discovery techniques. Data mining (i.e., knowledge discovery in database) is an IT artifact developed to aid human beings in addressing the information overload problem in knowledge discovery (Fayyad et al., 1996). Rooted in computer science, statistics, information science, cognitive science, etc., data mining research includes various tasks, such as classification, clustering, regression, and association rule learning. Traditional data mining research is usually conducted on data in relational databases, and has been extended in several directions according to the characteristics of the data. Examples include text mining research on free-text (Berry, 2004), Web mining research on files on the Web (Zhang and Segall, 2008), pattern recognition on multimedia data, temporal data mining (Roddick and Spiliopoulou, 1999) on data with time information, and spatial data mining (Roddick and Spiliopoulou, 1999) on data with location information, among others. In this dissertation, however, I will focus on knowledge discovery on graph-structured data, i.e., graph-based learning, and study its application in information systems.

1.2 Graph-structured Data

In this dissertation, I define graph-structured data as data instances (i.e., entities) that have relational information indicating their interactions or connections. While one relation connects a pair of entities, multiple connected entities constitute a network or a graph (in this dissertation, the two terms are used interchangeably unless specified). In the network, nodes are the entities being studied and links are the relations between the entities. As an example of graph-structured data, the social networks (Wasserman and Faust, 1994) in sociology and organizational science consist of individual persons linked by certain relationships, such as friendship. Webpages can also be viewed as a network, since they are linked by hyperlinks.

In this dissertation, I consider graph-structured data to be more than an abstract mathematical representation of nodes and links. The entities (nodes) may have rich features or data fields describing their attributes. The relations (links) may also contain information reflecting their semantics. In the example of social networks, each node (person) can be characterized by gender, career, age, etc. The links among individuals may be characterized by relation type (such as friendship, co-authorship, or mentorship), strength, sentiment (favorable or unfavorable), formation time, and so forth. In the example of Webpage networks, each node (Webpage) holds the rich textual information of its content. The hyperlinks can be annotated by the anchor texts, users' click through frequencies, etc., depending on the applications.

In graph-structured data, one network may contain more than one type of node or more than one type of link. In addition, the nodes/links can be either naturally occurring entities/relations in the applications or abstract concepts/implicit correlations extracted based on prior knowledge. For example, instead of building Webpage networks based on Webpages and hyperlinks, we can extract topic information from Webpages and build a topic-based network where nodes represent topics and links represent the similarity between topics. Such a network may be meaningful and has its utilities in knowledge mapping research.

Nowadays, graph-structured data can be found in various information systems applications (such as business intelligence, knowledge management, computer mediated communication, and medical informatics). Such examined data include friendship networks, co-authorship networks, company supply networks, document citation networks, customer-product purchase networks, customer-product review networks, and even gene interaction networks in medical informatics. There has been an increasing interest in using graph-structured data addressing business problems. However, traditional data mining methods have limited abilities to make use of the rich information embedded in networks. Thus, I devoted myself to studying knowledge discovery on graph-structured data in this dissertation.

1.3 A Framework for Graph-based Learning

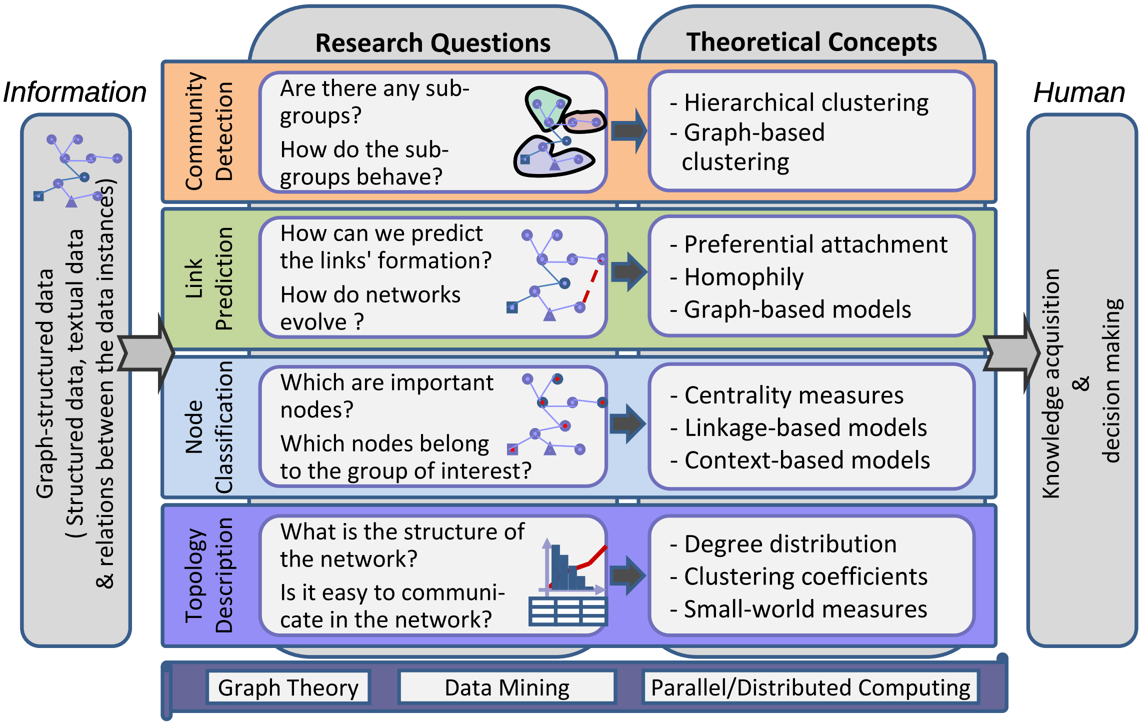

Research Questions
Theoretical Concepts
Information
Human
Graph-structured data
(Structured data, textual data
& relations between the data instances)
Community Detection
Are there any sub-groups?
How do the sub-groups behave?
- Hierarchical clustering
- Graph-based clustering
Link Prediction
How can we predict the links' formation?
How do networks evolve ?
- Preferential attachment
- Homophily
- Graph-based models
Node Classification
Which are important nodes?
Which nodes belong to the group of interest?
- Centrality measures
- Linkage-based models
- Context-based models
Topology Description
What is the structure of the network?
Is it easy to communi-cate in the network?
- Degree distribution
- Clustering coefficients
- Small-world measures
Knowledge acquisition
&
decision making
Graph Theory
Data Mining
Parallel/Distributed Computing

Figure 1.1 A Framework for Graph-based Learning

In this dissertation, I define graph-based learning as knowledge discovery on the basis of graph-structured data. I do not limit graph-based learning to any specific data mining/machine learn algorithm paradigms. However, I propose that such a graph-based learning process should take the graph structure into consideration. The graph representation highlights the relationships between data instances, especially indirect connections. It allows us to take advantage of graph theory developed in previous research into data mining for more effective knowledge discovery.

Figure 1.1 shows the graph-based learning framework prompted by this dissertation. The framework focuses on graph-structured data, i.e., structured/textual data and the relations between data instance, and targets at aiding human beings' knowledge acquisition and decision making processes. The framework is built upon data mining theory and graph theory. From the perspective of computational efficiency, the framework also considers parallel and distributed computing as part of the basis of graph-based learning. The framework contains four types of tasks for graph-based learning: topology description, node classification, link prediction, and community detection.

1) The topology description task aims to provide a global description of the network structure of the data. This task adopts the theoretical findings and statistical measures in social network analysis and graph theory to characterize the size and density of the network, the inter-connection patterns, positional relationships, and influences between nodes, etc. Such an analysis can help decision makers in the initial explorations of the data, which may lead to further more specific analyses.

2) The node classification task aims to group entities (nodes) according to predefined categories or criteria, such as classifying Webpages according to their topics. Such a task is similar to the traditional classification task, while the graph-based learning

framework emphasizes using not only nodes' local features but also relations between nodes to facilitate this task.

3) The link prediction task aims to infer possible links between nodes. It is similar to traditional association rule learning and relational learning studies, which are used to discover knowledge about entities' correlations. A network view enables us to explicitly use graph structure in the task and to analyze different types of relations (or relations between different types of nodes) in a unified framework. In addition, the predicted links also show expected network changes, which can aid in the global assessment of the network's evolution.

4) The community detection task aims to identify sub-groups of nodes according to their behaviors in the network. It is similar to traditional clustering analysis in the sense of grouping data instances. However, using a graph representation focuses more on the relationships between nodes in this task. In addition, the detected communities and their interconnection patterns provide us a mesoscopic description of the network. Community detection can be used to analyze the original network at a coarser level of granularity. It can also reduce computational requirements to facilitate our analysis of large scale networks.

I will further elaborate the proposed graph-based learning process for each one of the tasks mentioned above.

1.4 Summary of Dissertation Chapters

Chapter 2 focuses on the topology description task on the relational information extracted from documents, specifically gene interaction networks extracted from the biomedical literature and patent citation networks extracted from patents. This study empirically examines the characteristics of these networks, such as small-world and scale-free properties. In addition, this study compares the networks extracted from the documents with different methods and aggregated to different levels of analytical units using the network topology analysis measures. This chapter confirms the utility of network topology analysis in assessing a network's global characteristics in descriptive studies.

Chapter 3 and Chapter 4 examine the node classification problem with an example of patent classification and an example of gene function prediction, respectively. In Chapter 3, I propose to represent knowledge evolution processes using patent citation networks and model citation networks' structures for focal patents' classification. In a kernel-based framework, I propose a labeled graph kernel to capture the knowledge diffusion and evolution patterns on the patent citation network, which are related to the patents' topics. The approach complements traditional content analysis and significantly improves classification performance. In Chapter 4, I study the gene function prediction problem in the medical informatics domain by utilizing gene interaction networks. Based on a context assumption identified in previous gene function prediction research, I introduce a context graph kernel to capture features from context graphs, which includes the genes directly and indirectly interacting with the focal gene, for focal gene's function prediction. The approach achieves significantly better performance over traditional data mining methods. In addition, the study examines the mathematical formulation of the kernel and its performance characteristics under different parameter settings.

Chapter 5 showcases an example application of the link prediction task. In this chapter I frame the recommendation problem as a link prediction in a user-item purchase graph (which is a bipartite graph). I propose a one-class classification framework and a graph kernel model to tackle this problem by capturing the patterns in the structures of user-item pairs' associate interaction graphs. Such an approach provides more accurate predictions than prior methods, especially when a large number of recommendations for each user are needed. This study shows both the importance of the learning-based framework and the effective utilization of the graph-related features in addressing this type of difficult problem.

Chapter 6 studies the community detection problem in online environments. In this study I explore the use of link information, i.e., communication sentiments, in detecting online social groups. Based on an effective GN algorithm that can be applied on networks without sentiments, I design a GN-H co-training algorithm that uses both links with positive sentiments and links with negative sentiments in this task. Experiments on a simulated dataset show the superior performance of differentiating positive and negative sentiments. The experiments on an online product review dataset show the utility of the proposed method in aiding our analysis of online opinions.

In general, this dissertation exemplifies the effectiveness of utilizing graph structure information in the four knowledge discovery tasks using a variety of applications in business intelligence, knowledge management, computer mediated communication, and medical informatics. Its contributions to knowledge discovery and information systems are summarized in Chapter 7, which also presents future extensions of this work.

CHAPTER 2. TOPOLOGY DESCRIPTION: DIGESTING THE RELATIONAL INFORMATION FROM DOCUMENTS

2.1 Introduction

In this chapter, I introduce network topology analysis as a descriptive analysis approach to explore graph-structured data. Targeting the information overload problem in knowledge management, I analyze the relational information extracted from documents as an example application.

In recent years, the rapid development of modern technologies has led to a large increase in scientific literature and patent publications. For example, the number of new papers appearing in Medline rose from an average of 746/day in 1980 to 1,494/day by 2002 (Marshall et al., 2006). In the United States Patent and Trademark Office (USPTO), annual patent applications increased from 90,982 in 1963 to 417,508 in 2005 (Li et al., 2007b). Such large amounts of documents make it difficult for users to access information and for researchers to study and analyze the accumulation of knowledge and the diffusion of knowledge.

To address this problem, various analytical methods have been proposed for different information processing purposes. For the purpose of digesting the vast and growing collection of documents, text mining techniques have been developed to extract entities and relations from free text. In some application domains, such extracted relational information reflects the key knowledge elements embedded in the texts. For example, the knowledge from gene pathway research can be documented in biomedical documents as interactions between genes and their products. Another way of taking advantage of the collections of documents is to extrat inter-document relationships, which can facilitate analysis of knowledge diffusion patterns between knowledge holders. For example, the citation relationships between patents may indicate the transfer of knowledge elements from cited patents to citing patents.

Obviously, these two types of research make use of different types of relational information (intra-document relations and inter-document relations) and have significant semantic differences. However, the two types of information are all related to the accumulation and diffusion of knowledge within documents. Analyzing the two types of information may help us assess the landscape of knowledge development in application domains of interest. Since these relational data can all be represented as a network structure, network topology analysis methods can be adopted to assess their global characteristics which can be interpreted in their respective contexts.

In this chapter, I conduct two network topology analysis case studies. In the first case study, I compare the gene interaction networks extracted from biomedical literature by different information extraction techniques. In the second case study, I study the citation networks of USPTO nanotechnology patents aggregated to different levels of analytical units. In both case studies, network topology analysis methods enable us to unveil the global structural characteristics of the relational information accumulated in the literature and the knowledge diffusion patterns between different types of knowledge holders.

This chapter is structured as follows. Section 2.2 reviews previous research on network topology analysis. Section 2.3 describes the proposed framework for network topology analysis on relations extracted from documents. Sections 2.4 and 2.5 present the case studies on gene interaction networks extracted from biomedical literature and patent citation networks extracted from USPTO patents, respectively. Section 2.6 summarizes the findings.

2.2 Literature Review

In this section, I review previous network topology analysis studies in general. .Domain-specific literature will be reviewed in the two case study sections since there are different implications in their network topology analyses.

Network topology analysis employs various statistical measures to characterize the topology of complex networks. These measures describe important quantitative features such as the distance between nodes (average path length), tendency for the nodes to form clusters (clustering coefficient), node degree distribution, etc. Network topology analysis is also related to social network analysis studies in sociology. In 1967, Milgram discovered the six degrees of separation phenomenon (Milgram, 1967), which led later sociologists to study how social networks affect human behaviors.

In network topology analysis, three important graph models have been developed to understand the governing principles of network topology. 1) The Erdos-Renyi model or binomial model is a purely random model assuming links may appear randomly between nodes without any underlying governing principles (Erdos and Renyi, 1959). 2) The small-world model is a hybrid model that combines a regular lattice and a purely random graph to capture the co-existence of regularity and randomness (Watts and Strogatz, 1998). 3) The scale-free model incorporates the growth and preferential attachment mechanisms in networks (Barabasi and Albert, 1999). While the random graph model enables theoretical and numerical analysis as a baseline model, recent rich empirical literature on network topology analysis has found that the small-world property, characterized as the co-existence of short average path length and large clustering coefficient, and the scale-free property, characterized as having a power-law degree distribution, exist in a wide range of networks (Albert and Barabasi, 2002).

The network topology analysis methods are effective in describing network characteristics. They have been adopted to analyze different types of networks, such as the World Wide Web (Lawrence and Giles, 1998), social networks (Watts and Strogatz, 1998; Newman, 2001), biological networks (Jeong et al., 2000), telecommunication networks (Abello et al., 1999), and networks in linguistics (Ferrer et al., 2001). Among these studies, some of the relations naturally exist while the others need to be extracted from various data sources. In general, studies on the networks created from documents (such as scientific literature and patents) are limited.

2.3 Methodology

Based on previous network topology analysis studies, I propose a framework for analyzing relational information embedded in documents. The proposed methodology has four steps: document collection, relation extraction, network construction, and network topology analysis (Figure 2.1).

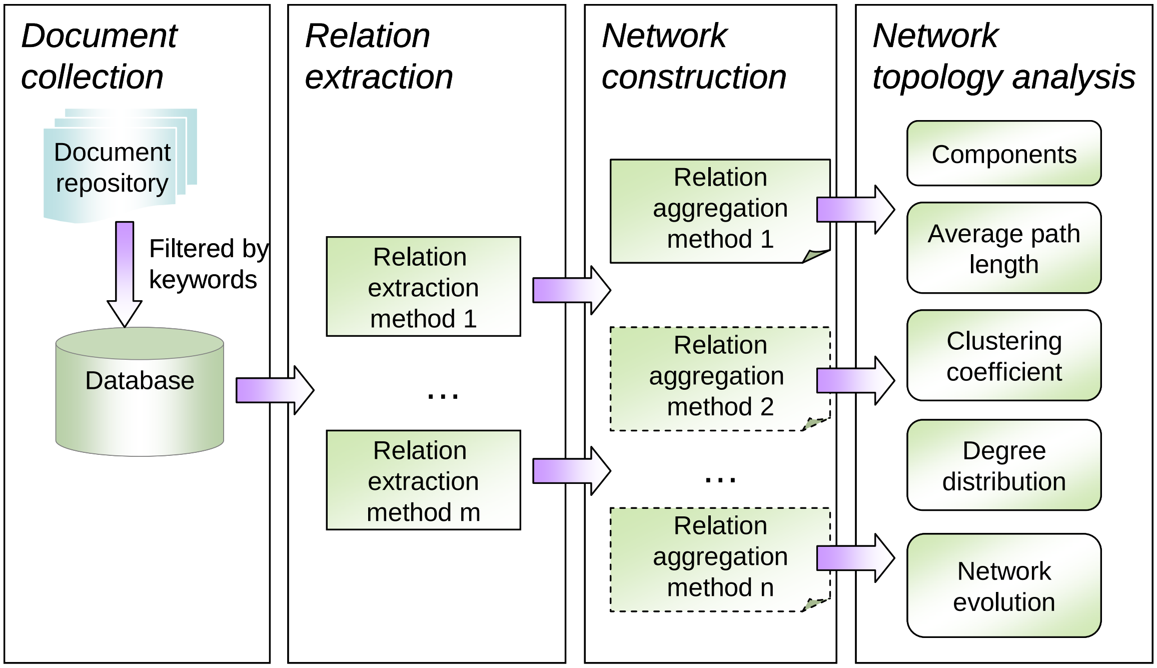


Figure 2.1 A Framework for Network Topology Analysis on Relations From Documents

1) Document collection

In the document collection stage, documents (papers, patents, etc.) need to be extracted from relevant domain repositories (such as Medline and USTPO). I propose to take a keyword search approach for this procedure, since keyword search is a functionality provided by most repositories. Depending on the purpose of the study, the keywords can be used to match the full-text of the documents or some of the data fields of the documents. The extracted documents usually need to be parsed and saved in structured databases.

2) Relation extraction

In the relation extraction stage, the relational information of interest to the users needs to be extracted. In general, accurately extracting relational information from document contents needs significant parsing efforts or Natural Language Processing

(NLP) techniques. It can be relatively easier to extract inter-document relations such as citation relations. However, the various reference standards and noise in the references may still need to be addressed. In short, in this stage, various relation extraction tools/techniques can be adopted.

3) Network construction

In the network construction stage, the extracted relations are aggregated together to build the network. In this research, I only consider unipartite networks which contain one type of nodes. Thus, all entities connected by extracted relations need to be aggregated to a certain type of analytical units. For example, we can aggregate biological relations to the gene level; we can also aggregate patent citations to the country level. When the entities are aggregated to a certain type of nodes, the links between two nodes represent all relations between the entities of the two groups. Thus, one can weight a link with the total number of relations represented by the link. However, in this study I consider only unweighted networks. Depending on the nature of the relations, the links can be directional or non-directional.

4) Network topology analysis

In the network topology analysis stage, various statistical measures are employed to characterize the topology of created networks (Albert and Barabasi, 2002):

a) Network size: The number of nodes (*Node#*) and links (*Link#*) in the network represents the coverage of the relations.

b) Component size: A component is an isolated sub-network in a disconnected network. The number of components (N_C), number of nodes of the giant component ($Node\#_C$), and number of links of the giant component ($Link\#_C$) are used to characterize the components.

c) Network diameter (D): The maximum value of the shortest path length between any pair of nodes in the network.

d) Average path length (l): The average value of the shortest path length between any pair of nodes in the network.

e) Clustering coefficient (C): The network's clustering coefficient is the average of each node's clustering coefficient *C'*. A node's clustering coefficient is the ratio of the number of edges between the node's neighbors to the number of possible edges between those neighbors (one node's neighbors are the nodes directly connected to it):

$$C' = \frac{\text{number of edges between the neighbors}}{\text{possible number of edges between the neighbors}}.$$

f) Average degree ($<k>$): The average number of links that a node has to other nodes.

g) Degree distribution $P(k)$: Degree distribution represents the probability that a selected node has exactly k links:

$$P(k) = \frac{N(k)}{N},$$

where $N(k)$ is the number of nodes with k links and N is the total number of nodes. It should be noted that in a directional network, a link has a start node and an end node. The in-degree of a node is the number of links that have the node as an end. The out-degree of a node is the number of links that have the node as a start. The degree distribution can be calculated based on both in-degree and out-degree.

h) Network evolution: If the relations of a network are stamped with the time of creation, the entire network can be sliced to sub-networks according to the relations' time of creation. The temporal changes of the topological measures of this series of sub-networks show the evolution of the relations in documents. It is also possible to test some network evolution models on the dataset, such as the preferential attachment model for scale-free networks (Barabasi and Albert, 1999; Jeong et al., 2003).

2.4 Case Study 1: Gene Interaction Networks Extracted From Literature

2.4.1 Background

Genetic interactions control many important biological processes in cells. Traditionally, characterization of individual gene regulatory pathways was the focus of genomic research. Recent studies have switched to constructing and analyzing networks at a genome-wide scale to assess the global characteristics of genetic interactions (Barabasi and Oltvai, 2004).

Gene interactions can be extracted from high-throughout experimental data, such as microarray (Luscombe et al., 2004; Tong et al., 2004), data from mass spectrometric analysis (Gavin et al., 2002; Ho et al., 2002), and two-hybrid screening (Jeong et al., 2001; Yook et al., 2004), using a variety of analytical methods.

Biomedical literature provides another source of gene interactions, which are human knowledge from previous research. Gene interactions can be manually curated by domain experts based on previous research and literature (Fell and Wagner, 2000; Shen-Orr et al., 2002; Ma and Zeng, 2003). The recent advances in text mining techniques make it possible to automatically extract gene/protein interactions from biomedical literature. Co-occurrence analysis identifies the entity pairs that appear in the same context (Stapley and Benoit, 2000; Jenssen et al., 2001). Although not every co-occurrence relation reflects an actual interaction between the two genes, statistically significant co-occurrence relations based on a large corpus of literature may correspond to underlying gene interactions (Wren et al., 2004). Parsing relations using Natural Language Processing (NLP) technology is another approach to gene/protein interaction extraction (McDonald et al., 2004; Marshall et al., 2006). Chen and Sharp developed a system which incorporates NLP tools to parse syntactic gene relations from Medline abstracts and reported the degree distribution of some parsed relation network examples (Chen and Sharp, 2004).

In previous gene interaction network studies, both small-world and scale-free characteristics were found in gene interaction networks (Shaw, 2003; Tari et al., 2005) and protein interaction networks (Jeong et al., 2001; Yook et al., 2004). In addition, several studies discovered that network motifs, i.e., recurrent interconnection patterns in local structures, exist in gene interaction networks (Shen-Orr et al., 2002; Luscombe et al., 2004) and protein interaction networks (Wuchty et al., 2003). However, most previous gene interaction network studies were conducted based on experimental data and manually curated data. The use of relations automatically extracted from a large body of biomedical literature using modern text mining techniques is limited. Experimental data are usually limited to particular experimental conditions. Manually curated data usually require intensive labor by domain experts. Conducting topology analysis on automatically extracted relations from biomedical literature may provide us a better idea of the cumulative human knowledge of gene interactions.

2.4.2 Dataset

To test the proposed framework, I created a p53-related testbed from Medline. P53 is a tumor suppressor gene playing a central role in cancer development, which is of interest to many biologists. The p53-related documents were collected from Medline by matching the abstracts that contain various names of p53 and other genes in the p53 pathways, as suggested by domain experts. In total, 87,903 abstracts related to p53 (1975-2003) were identified.

I leveraged previously developed text mining techniques to extract the gene interactions from the collected Medline abstracts. The process consists of two major steps: parsing gene/protein relations with the Arizona Relation Parser (McDonald et al., 2004) and aggregating the parsed relations with the BioAggregate tagger (Marshall et al., 2005). In the first step 51,033 distinct entities and 44,864 relational triples were extracted. The parsed relations were aggregated into 8,837 genes and 29,635 gene interactions in the second step.

I constructed two types of gene interaction networks from the extracted relations: a parsed network and two co-occurrence networks. To construct the parsed network, I kept only the aggregated relations with genes on both sides, which contain 2,045 genes and 6,092 interactions. Each aggregated gene interaction has a time tag which indicates the first time it was documented in biomedical literature. For the co-occurrence networks, I first generated a co-occurrence network on the gene pairs that co-occurred in at least one abstract, which contains 4,233 genes and 33,968 relations (some abstracts only contain one gene, which were removed in this procedure). As co-occurrence relations appearing multiple times are more meaningful, I created a reduced co-occurrence network by only including the co-occurrence relationships that appeared in two or more abstracts to reduce the network to the same scale as the parsed network. The reduced co-occurrence network contains 2,017 genes and 10,104 relations.

2.4.3 Results and Discussion

2.4.3.1 Topological Measures

Table 2.1 Topological Measures of the Gene Interaction Networks

	Parsed Network	Co-occurrence Network	Reduced Co-occurrence Network
Node#	2,045	4,233	2,017
Link#	6,092	33,968	10,104
$<k>$	5.958	16.050	10.019
l	3.318	2.884	2.891
l_{rand}	4.271	3.009	3.302
C	0.3149	0.6254	0.6769
C_{rand}	0.0029	0.0038	0.0049
D	8	8	8
N_C	37	51	30
$Node\#_C$	1,967	4,125	1,956
$Link\#_C$	6,050	33,903	10,071

The topological measures of the parsed network, co-occurrence network, and reduced co-occurrence network are shown in Table 2.1. All three networks are composed of several components. They all have a giant component which has most of the genes in the network. For example, the giant component of the parsed network contains 96% (1,967/2,045) of the nodes and 99% (6,050/6,092) of the links. The existence of the giant components, which is also found in other biological networks (Ma and Zeng, 2003), indicates a high degree of interdependency between the genes involved in cellular processes.

Table 2.1 shows that all three networks have a large clustering coefficient and a small average path length compared to random networks of the same size. For example, the parsed network has a much larger clustering coefficient (0.3149) and a smaller average path length (3.318) than those of a same-size random network (0.0029 and 4.271, respectively). These properties reflect the small-world characteristics of the networks. A small average path length indicates that one gene's effect can be quickly propagated to other genes in the biological process. A large clustering coefficient indicates that the genes interacting with one gene tend to interact among themselves as well. In other words, there is a probability of the existence of local clusters.

There is a major difference in the size of the reduced co-occurrence network and the co-occurrence network, but their average path length and clustering coefficient are quite similar. In the co-occurrence network genes appearing in the same abstract form a fully connected cluster and the network is made up of those local clusters. Thus the co-occurrence network has a high clustering coefficient and a small average path length. The reduced co-occurrence network is formed by removing weak co-occurrence relations that only occurred in one abstract, which might not represent an actual gene interaction relationship. The similarity in the topological measures of the two networks indicates that removing the rare co-occurrence relations from the network does not substantially change the network topology.

Although the three networks have similar average path lengths, there is a large difference in their clustering coefficients. The clustering coefficients of the reduced co-occurrence network (0.6769) and the co-occurrence network (0.6254) are about twice as large as that of the parsed network (0.3149). The substantial difference in clustering coefficients reflects the nature of the three different networks in local cluster (highly connected sub-graph) formation. Comparing the reduced co-occurrence network and the parsed network, we can see that although the two networks have similar numbers of genes, the reduced co-occurrence network has a much larger clustering coefficient. This indicates that the reduced co-occurrence network captured many more relations and has more significant local clusters.

2.4.3.2 Degree Distribution

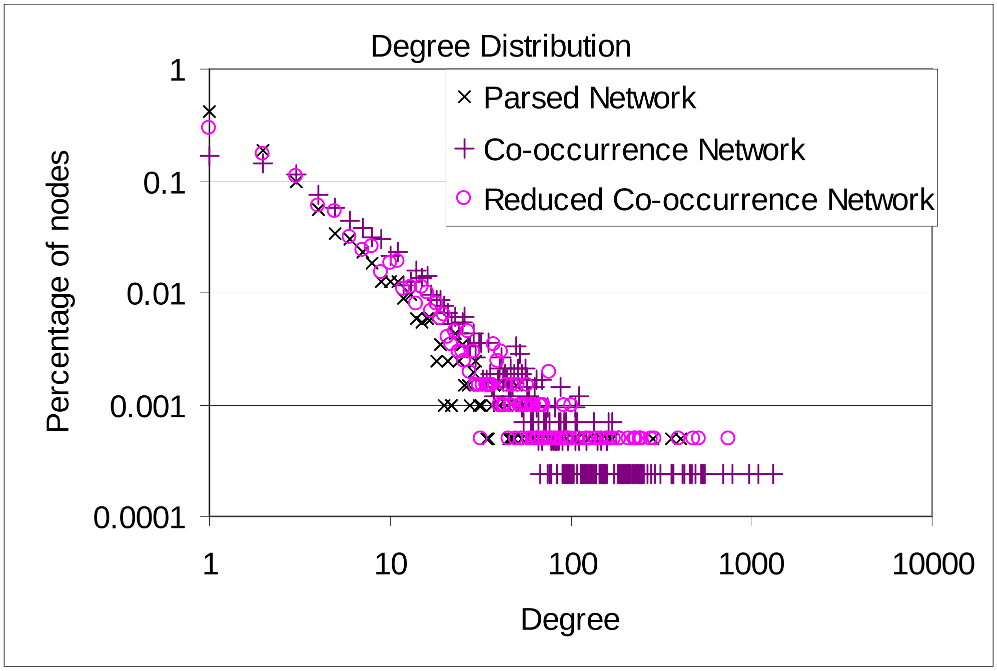


Figure 2.2 Degree Distribution of the Gene Interaction Networks

Figure 2.2 shows the degree distributions of the three networks, which are close to a straight line on the log-log plot, indicating that they follow a power-law distribution. A power-law distribution means that the number of nodes with a certain degree in the network decreases quickly when the degree increases. The scale-free characteristics of the networks in the research may have two causes. 1) The actual gene interactions in biological processes follow the power-law distribution. 2) As the data is on the discovered genes and relations in the literature, the scale-free characteristics may be a result of the collective knowledge creation and accumulation process of human beings—researchers tend to conduct research related to the known important genes.

2.4.3.3 Network Evolution

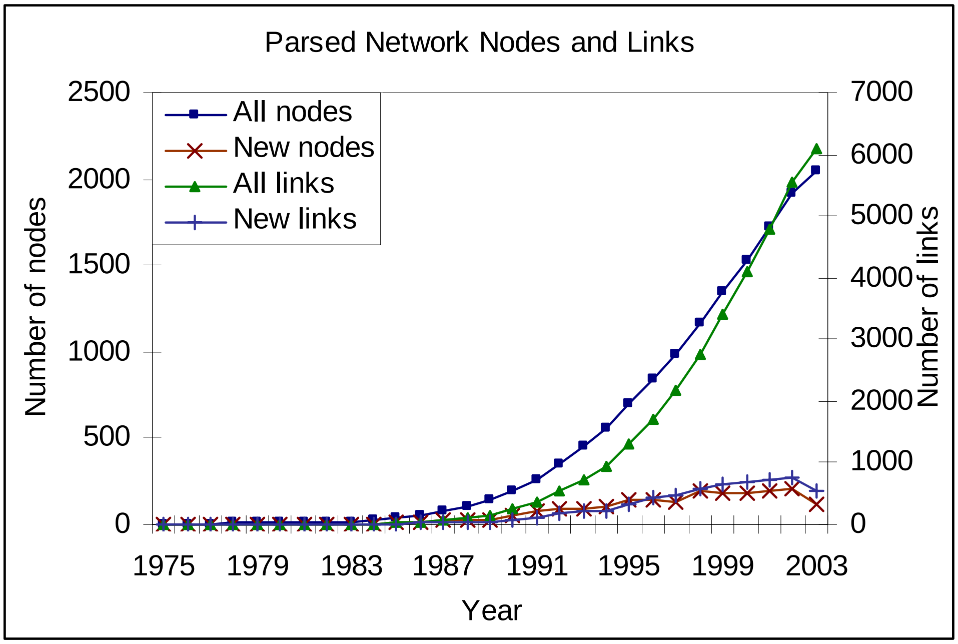


Figure 2.3 Evolution of the Parsed Network

Since the extracted gene interactions in the parsed network are associated with timestamps, it is possible to analyze the evolution of this network of gene interactions.

Figure 2.3 shows the evolution of the number of nodes and links of the parsed network. There is a consistent growth in the number of new interactions and genes, especially in recent years (after 1991). The decrease in the number of newly discovered gene interactions in 2003 is because of incomplete data at the time of this study. Except for that, there is no indication of the network's convergence to a fixed set of genes. It is possible that more genes involved in the p53 pathway will be identified.

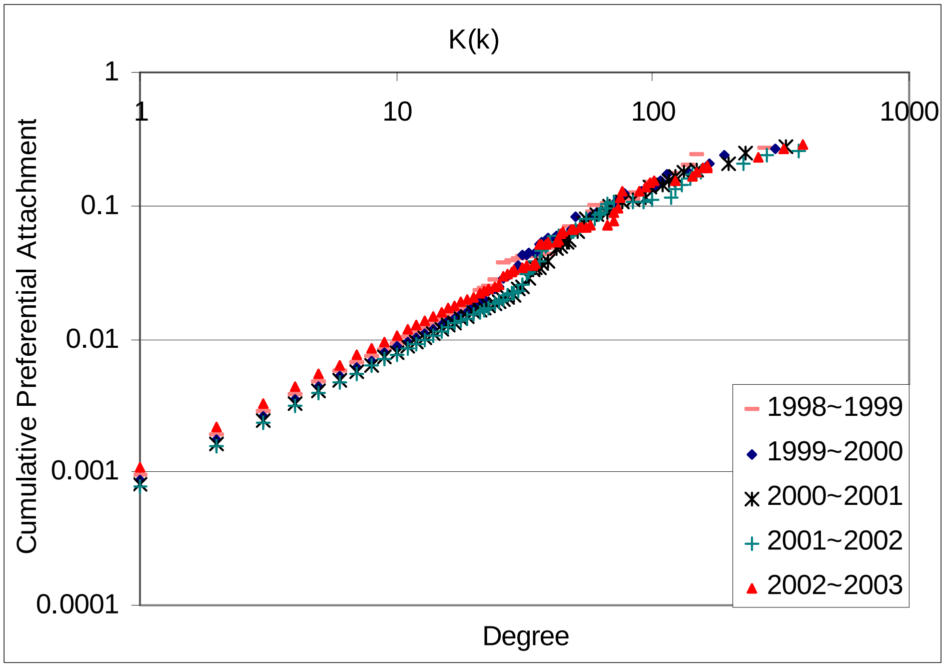


Figure 2.4 Preferential Attachment Test

In Figure 2.4, the preferential attachment test (Jeong et al., 2003) shows more details about the expansion of the parsed network. The preferential attachment tests for the years from 1976 to 2003 follow similar patterns. I only report results for the most recent five years to make the graph easier to read. The straight line of cumulative preferential attachment *K(k)* with positive slope in the log-log graph means that node *i*'s probability to get new links *$P_i(k)$* is proportional to its degree *k*. Thus, the probability a gene will be found to interact with other genes is proportional to the number of known interactions involving it. This indicates that researchers tend to focus more on the well-studied genes and study the gene interactions related to them. This analysis shows evidence that the nature of the collective research exploration process at least partially accounts for the observed power-law degree distribution of the literature-based networks reported earlier.

2.5 Case Study 2: Patent Citation Networks

2.5.1 Background

Patents contain significant amounts of knowledge on technology innovations. The analysis of knowledge in patents has been of interest to researchers for years. Patent publication has been used as one major indicator of research productivity and impact (Narin, 1994). For example, previous research has used patents to analyze the longitudinal change of the international landscape of nanotechnology research and development (Huang et al., 2003b) and the development of high-tech electronics companies in Taiwan (Huang et al., 2003a).

Patent citation information has been used to represent knowledge transfer (Karki, 1997; Oppenheim, 2000). Chakrabarti et al. analyzed the inter-organization patent citation patterns of defense-related research and development on the civilian sector (Chakrabarti et al., 1993). Chen and Hicks studied the interactions between academia and industry by analyzing the paper-patent citations in the field of tissue engineering (Chen and Hicks, 2004). These studies were usually based on the citation patterns between entity pairs. Performance measures based on patent citation (e.g., number of cites of a patent or an assignee) were used to describe such “local” characteristics of knowledge conveyance.

Recent advances in network topology analysis methods can enhance our understanding of patents’ knowledge flow by considering a network view of patent citations. Topology analysis has been incorporated into recent literature/patent citation network studies. It was found that the literature citation network is a tree-like network and a scale-free network (Bilke and Peterson, 2001) with a power-law degree distribution (Redner, 1998). The power-law degree distribution phenomenon was also found in patent citation networks (Chen and Hicks, 2004). In this case study, I apply network topology analysis methods to explore the knowledge diffusion patterns in patents at different analytical unit levels, including document level, country level, institution level, and technology field level.

2.5.2 Dataset

I chose nanotechnology as the application domain of this research due to its internationally-recognized importance. I used a nanotechnology-related keyword list provided by domain experts (Huang et al., 2003b; Huang et al., 2004a) and conducted “full-text” search (matching the keywords in the patent title, abstract, claims, and description) to collect nanotechnology patents from the USPTO database. After filtering out the patents that matched keyword patterns but were not related to nanotechnology research (such as the patents that only contain keywords “nanosecond” or “nanometer”), 78,609 patents published between 1976 and 2004 were identified. These patents were filed by 22,219 assignees institutions and 163 assignee countries. They cover 432 of the

462 first-level United States Patent Classification (USPC) categories, which were used to represent patents' technology fields in this research.

The patent citation relations are easy to parse from patent documents. First, the citation relations were constructed into a document level patent network. Then, the relations were aggregated together according to patents' assignee country, assignee institution, and technology fields to create corresponding networks at different analytical unit levels. During this process, the patents that were not involved in any citations were removed. The resulting document level network contains 54,730 patents and 140,872 citation relations; the country level network contains 59 countries and 423 aggregated relations; the institution level network contains 10,878 institutions and 44,828 aggregated relations; and the technology field level network contains 397 technology fields and 14,487 aggregated relations.

2.5.3 Results and Discussion

2.5.3.1 Topological Measures

Table 2.2 reports the topological measures of the patent citation networks at document, country, institution, and technology field level. The document level citation network contains 2,969 components, among which the biggest component contains 45,717 patents and 133,769 relations. The patent document citation network has a much larger average path length (l = 8.923) than the random network of the same size (6.658), which is different from most large scale networks (Albert and Barabasi, 2002). The knowledge transferring process in this network is not as efficient as that in a random network. In a random network, links may appear between any pair of nodes. In a patent document citation network, one patent may only cite its related documents, which may cause this phenomenon. The patent document citation network shows a very large clustering coefficient (C = 0.1781) compared with the random network of the same size (9.41E-05). The high clustering tendency of patents directly follows from the nature of citations. Two patents with a citation relationship often involve similar technology and are highly likely to be cited together by later patents. Such close citation relations may create the local citation clusters and increase the clustering coefficient.

Table 2.2 Topological Measures of the Patent Citation Networks

	Document Level	Country Level	Institution Level	Technology Field Level
Node#	54,730	59	10,878	397
Link#	140,872	423	44,828	14,487
$<k>$	5.147	8.305	7.571	58.317
L	8.923	1.933	3.754	2.007
l_{rand}	6.658	1.926	4.591	1.472
C	0.1781	0.841	0.3342	0.7168
C_{rand}	9.41E-05	0.1432	0.0007	0.4907
D	36	4	15	6
N_C	2,969	1	352	3
Node#$_C$	45,717	59	10,220	395
Link#$_C$	133,769	423	39,770	14,485

The country citation network represents the patent citation relationship at the assignee country level. It only contains one network component. In other words, every country in the network directly or indirectly affects the other countries through the published patents in nanotechnology research. The country citation network has a small diameter (D = 4) and a small average path length (l = 1.933). The average path length of the country citation network is close to that of a same-size random network (1.926). Thus, the knowledge diffusion process in the country citation network is as effective as that in a randomly connected citation network. The country citation network has a larger clustering coefficient (C = 0.8410) than that of the random network of the same size (0.1432). The large clustering coefficient means that the countries that have citation relations to a particular country may have a high tendency to interact with each other. For example, some of the major countries in the network, including the United States, Japan, Germany, and the United Kingdom, form a cluster with close citations.

The institution citation network represents the patent citation relationships at the assignee institution level. The institution citation network consists of 352 components. The largest component contains 10,220 institutions and 39,770 citation relations. The institution citation network has a smaller average path length (l = 3.754) than the random network of the same size (4.591). The institutions in the institution citation network are connected by fewer steps of citations as compared with a random network. Knowledge can be transferred efficiently from the major inventors, such as IBM, the University of California, and 3M Company, to other institutions. The institution citation network has a much larger clustering coefficient (C = 0.3342) than the random network (0.0007). Institutions have a tendency to form local citation clusters because of similar interests and collaborations. From the clustering coefficient measure, we can infer that intensive knowledge transfer among groups of peer institutions is very common.

The nanotechnology technology field citation network has three components. The largest component contains 395 of the technology fields and 14,485 of the citation relations. The technology field citation network has a very high average degree ($<k>$ =

58.317) with only 397 nodes. It is much higher than the average degree of the country citation network and the institution citation network. The high average degree shows the close, interacting relationship among the technology fields, suggesting that nanotechnology is a highly interdisciplinary domain. The technology field citation network has a small average path length (*l* = 2.007) and a small diameter (*D* = 6). On average, the knowledge in one technology field can be transferred to the others through two steps of citation relations. However, the average path length is still large in comparison with the random network of the same size (1.472). This means that although knowledge could transfer quickly in the technology field citation network, it is slower than the knowledge transfer process in the random citation network. The technology field citation network is a dense network with a high clustering coefficient (C = 0.7168), which is larger than that of a random network of the same size (0.4907). This network does not show a distinct local cluster characteristic compared with the random network.

2.5.3.2 Degree Distribution

Because the weight (number of citations) was not reflected in the degree distribution, I did not study the degree distributions of the country citation network, the institution citation network, and the technology field citation network, which are all weighted networks.

Figure 2.5 shows the in-degree distribution and out-degree distribution of the document level citation network. The out-degree distribution shows the probability of the number of citations one patent may receive. The in-degree distribution shows the probability of the number of references one patent may have. In the log-log graph, the two degree distributions show the pattern of a straight line, which means that they follow the power-law distribution. The power-law distribution takes the form of $P(k) \sim k^{-\gamma}$, where $P(k)$ is the probability that a node has exactly k links. The power-law exponent γ and correlation coefficient r of the two degree distributions are shown in Table 2.3. The power-law degree distribution shows that the patent document citation network follows the scale-free model, which indicates that a few high degree patents exist in the network. The high out-degree patents represent critical, far-reaching, fundamental innovations that influence many other patents.

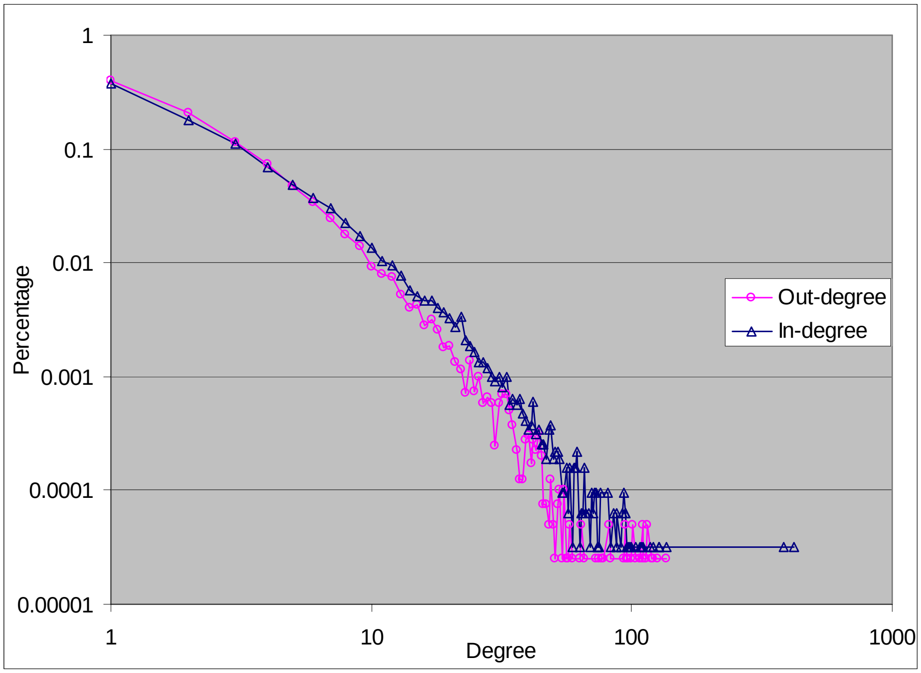


Figure 2.5 Patent Citation Network Degree Distributions: In-degree and Out-degree

Table 2.3 Patent Document Citation Network Degree Distribution Measures

	power-law exponent γ	correlation coefficient r
Out-degree distribution	2.2925	0.6855
In-degree distribution	2.1394	0.8528

2.6 Summary

In this chapter, I adopted a network topology analysis framework to conduct two case studies on relational information extracted from documents. The first study analyzes the gene interaction networks extracted from biomedical literature, which represent the biological knowledge accumulated on gene pathway studies. The second study analyzes patent citation networks in the nanotechnology domain, which represent the knowledge diffusion patterns in that field. The study compared the networks constructed with different text mining techniques and the networks aggregated to different analytical unit levels, respectively.

In the first case study, I created a testbed of Medline abstracts related to p53 pathways. In the p53 dataset, the gene interaction networks extracted from literature using the NLP approach (parsed network) and using the co-occurrence approach (reduced co-occurrence network) show similar topological characteristics. These networks all have small-world and scale-free properties. Comparison of the networks shows that the reduced co-occurrence network contains more significant local clusters than the parsed network, while the parsed network contains less noise than the reduced co-occurrence network. The evolution of the parsed network shows preferential attachment characteristics, which is consistent with other large-scale networks.

In the second case study, I created a testbed of nanotechnology patents from the USPTO database. I created citation networks among them at different analytical unit levels (i.e., document, country, institution, and technology field). Network topology analysis shows that all four citation networks have one large “giant” component that occupies most of the nodes and links. The four networks have different knowledge transfer efficiency as compared to a random network of the same size. According to the average path length measure, the institution citation network structure exhibits a more efficient knowledge transfer than a random network. The country citation network shows a knowledge transfer capability as efficient as a random network. The technology field citation network and the patent document citation network have a less efficient knowledge diffusion capability than the random network. According to the clustering coefficient measure, the country citation network, the institution citation network, and the patent document citation network all show a tendency to form local citation clusters, which indicates the intensive cooperation and knowledge exchange between these analytical units.

With these two case studies, I have showcased the use of network topology analysis methods in descriptive analysis. The framework proposed in this research is a general framework that can be applied to relations extracted from other types of documents. The two case studies on gene interaction networks and patent citation networks confirmed that complex networks are not random. The small-world and scale-free phenomenon indicates the rich information embedded in the network structures. In the next chapter, I will explore the use of this network structure information in addressing the node classification tasks.

CHAPTER 3. NODE CLASSIFICATION: TRACING KNOWLEDGE EVOLUTION WITH CITATION NETWORKS FOR PATENT CLASSIFICATION

3.1 Introduction

In this chapter, I explore the use of network structure information in graph-structured data to address the node classification problem. I use a critical knowledge management problem, patent classification, as an example application of the node classification task.

Due to the information overload problem, managers face more challenges in organizing and managing knowledge for future sharing and usage (Nidumolu et al., 2001). To facilitate the knowledge management (KM) tasks, automated tools have been widely adopted (Spangler et al., 2003). However, most available tools treat knowledge items independently and only process their contents.

Knowledge items are not independent from each other. Knowledge evolves after transfer and reuse in human collaborations (Bieber et al., 2002). Knowledge creation has been considered as a path-dependent evolution process (Nerkar, 2003), where innovation is created based on the recombination of prior knowledge elements (Fleming, 2001). From this perspective, the knowledge evolution processes may affect the newly created knowledge and should be taken into account in KM tasks. The knowledge evolution process can be embedded in the relationships among individual documents, such as patents and scientific literature citations. In this research I choose one type of such “linked document,” patents, and conduct an empirical study to exploit the utility of knowledge evolution processes in KM tasks.

Patents contain a significant amount of knowledge on technical innovations. Patent management, at both the organization level and the society level, prompts the exchange of inventions (Scherer, 2002) and reduces the duplication of research efforts (Gallini, 2002). Due to the surge of patent applications and publications in recent years, patent processing time has been significantly prolonged (Hunt, 2001) while the patent examiners' workload has been continuously increasing (King, 2003), which hinders effective patent management and affects inventors' benefits. Classification plays a critical role in patent management , including assigning patent applications to examiners (Smith, 2002) and organizing patents based on patent classification schemes, e.g., the United States Patent Classification (USPC) system. Improving patent classification performance may affect the efficiency of patent examination and the effectiveness of patent search systems.

Most previous studies on patent classification focused on content analysis and treated the problem as a canonical text categorization problem (Sebastiani, 2002; Loh et al., 2006). Although various features extracted from patent contents have been used and several machine learning algorithms have been applied (Fall et al., 2003), such approaches have not provided satisfactory performance (Smith, 2002).

In order to utilize knowledge evolution processes in the patent classification task, I use patent citations to represent the knowledge diffusion and reuse processes in innovation creation (Almeida and Kogut, 1999; Singh, 2005), in the sense that citing patents adopt knowledge elements from cited patents. Under a kernel-based machine learning framework, I explore different methods to model patent citation networks, which capture knowledge evolution processes, to facilitate patent classification. I propose a novel model, the labeled graph kernel, which shows a significant improvement in classification performance as compared with traditional content-based approaches. I also identify both the citation network structure and the features of cited patents as important factors in describing knowledge evolution processes for patent classification. This study shows the possibilities for further automating the patent examination process and the benefits of considering the knowledge evolution process in KM tasks.

This chapter is structured as follows. Section 3.2 reviews previous research on patent classification in the context of linked document classification. In this section, I also briefly review kernel-based methods on structured information. Section 3.3 describes a kernel-based approach and proposes several kernels that use patent citation networks and patent contents for classification. Section 3.4 reports the experiments on a nanotechnology-related patent testbed. Section 3.5 discusses the experimental results. Section 3.6 summarizes the findings.

3.2 Literature Review

As a common type of knowledge, linked documents such as patents, scientific literature, and Webpages are associated by links in the form of citations or hyperlinks. From a knowledge management perspective, the document content contains different forms of knowledge, while the links among them indicate the process of knowledge transfer and diffusion.

The classification of linked documents is of interest to both managers and scholars. Classification tools have been developed and adopted in patent management

(Smith, 2002), Webpage management (Craven and Slattery, 2001; Furnkranz, 2002) and scientific literature management (Spangler et al., 2003; Ginsparg et al., 2004; Sinclair and Webber, 2004). Among these tasks, patent classification has its unique challenges due to its critical role in practice and its data characteristics (Smith, 2002). Patent classification is usually conducted on a large number of categories (for example, the USPC has 450 first-level categories and 160,000 second-level categories). Many of these fine-grained classes have subtle semantic differences and usually have an uneven number of patent instances (Krier and Zacca, 2002). All these factors make patent classification difficult to address compared to other linked document classification tasks.

3.2.1 Classification of Linked Documents

I review previous patent classification studies in the context of linked document classification from two aspects: features, i.e., how the documents are represented, and algorithms, i.e., how the documents are classified.

3.2.1.1 Feature Types

Previous studies on the classification of patents mainly consider the features in individual documents. Features related to the citations (links) between documents have also been used.

1) Features of individual documents:

Most previous research considered only the knowledge embedded in individual patents and extracted features from individual documents to represent patents. These features can be categorized into content features and metadata features. Content features are often considered good indicators of document subjects, which can be extracted at the word level (i.e., “bag-of-words”) or phrase level (Ghanem et al., 2002) from different parts of the documents. In patent classification, previous studies examined the features extracted from patent title (Larkey, 1999), abstract (Larkey, 1999; Fall et al., 2004; Loh et al., 2006), claims (Hull et al., 2001), and full-text (Koster et al., 2003). Features from patent title and abstract have been found to be more effective in patent classification.

The metadata, which usually describe the document’s author, institution, publication date, etc., may be highly correlated with its content and topic. In patent classification, Richter and MacFarlane have used a patent’s International Patent Classification (IPC) category to help classify it into another classification scheme (Richter and MacFarlane, 2005). In Webpage classification, Yang et al. used Webpage headers to help label Webpages by industry sectors (Yang et al., 2002). These studies demonstrated metadata’s effectiveness in improving classification performance.

2) Features of citations/links:

In machine learning literature, citations (links) indicate the close relationship between linked documents’ topics, methods, etc. From the knowledge creation perspective, citations (links) indicate the inheritance or transfer of knowledge elements between linked documents (Fleming, 2001). In linked document classification, features can be defined on direct citations or the entire citation network (of directly and indirectly connected documents) by considering different levels of the knowledge evolution process.

The simplest way to take advantage of direct citations is to combine features of the neighboring (directly cited) documents and use them to describe the focal document. Studies in both patent classification (Chakrabarti et al., 1998) and Webpage classification

(Oh et al., 2000; Ghani et al., 2001; Yang et al., 2002) have shown that combining the neighbor documents' content features cannot significantly improve classification performance. However, it has been found that combining neighbor documents' classification category (metadata) features does yield improvement (Chakrabarti et al., 1998; Oh et al., 2000).

Another method that utilizes direct citation information is to define features on linkage relationships. In Webpage classification research, hyperlinks have been represented as first-order logic clauses to build first-order rules describing the common characteristics of Webpages in the same category (Slattery and Craven, 1998; Craven et al., 2000; Craven and Slattery, 2001; Yang et al., 2002). Document similarity measures based on document in-links (co-citation similarity) (Small, 1974), out-links (bibliographic coupling similarity) (Kessler, 1963), or both in-links and out-links (Amsler similarity) (Amsler, 1972) have been used with the K-nearest neighbor (KNN) algorithm and the Support Vector Machine (SVM) algorithm (Joachims et al., 2001; Cristo et al., 2003; Calado et al., 2006) in both Webpage and scientific literature classification studies. Although citation measures have been widely used in patent analysis studies to assess the impact of patents, inventors, and assignees (Narin, 1994; Huang et al., 2003b), few previous studies have taken advantage of linkage features.

While using direct citations only considers a single step of the knowledge transfer between citing and cited documents, using features extracted from the entire citation network is a natural extension that gives a more complete picture of the knowledge evolution process. In recent studies on network topological analysis, researchers found that the networks of patents (Li et al., 2007a), Webpages (Broder et al., 2000), and scientific literature (Redner, 1998) are different from random networks. Their organized topological characteristics indicate rich information is contained in these networks. However, few studies have considered using features defined on patent citation networks to represent the knowledge transfer and innovation generation processes in patents and to address the patent classification problem.

3.2.1.2 Algorithm Types

The algorithms used in patent and other linked document classification can be categorized into feature-based methods and kernel-based methods.

1) Feature-based methods:

Feature-based methods are the major approach used in previous patent classification research. In feature-based methods, a data instance is represented by a feature vector, in which the features are explicitly constructed and selected based on domain knowledge or using automatic algorithms. In patent classification, KNN (Teichert and Mittermayer, 2002), Winnow (Krier and Zacca, 2002; Koster et al., 2003), Naïve Bayes, and probabilistic relational model (PRM) (Taskar et al., 2002) have been widely applied on content features. Feature-based methods can utilize different types of information by incorporating different types of features in the feature vector. In previous research, content features and neighbor document features (direct citation features) have been used together with the Naïve Bayes algorithm in both patent and Webpage classification (Chakrabarti et al., 1998; Oh et al., 2000).

2) Kernel-based methods:

Unlike feature-based methods, kernel-based methods do not require the explicit definition of feature vectors. A kernel-based method contains a kernel function and a kernel machine. The kernel function (or kernel) maps data instances from the input space χ to a feature space H (named reproducing kernel Hilbert space, RKHS) $\Phi(x):\chi \to H$, by defining a similarity measure between data instances $k:\chi\times\chi\to\Re$ $(x,x')\to k(x,x')$. Although $\Phi(x)$ is not explicitly defined, for every pair of data instances the kernel function ensures that $k(x,x')=<\Phi(x),\Phi(x')>$. A kernel machine, such as SVM, is a learning algorithm which performs learning tasks in the feature space *H* (Gartner, 2003). Given limited types of kernel machines (with SVM being state-of-the-art), the performance of kernel-based learning is highly dependent on the selection and design of kernel functions (Tan and Wang, 2004).

In linked document classification, kernel-based methods have not been used as widely as feature-based methods. However, they have shown their potential in some recent studies. For example, Fall et al. compared the performances of KNN, Naïve Bayes, and Winnow with SVM on a linear kernel using content features and found that SVM with the linear kernel outperformed the other three feature-based methods (Fall et al., 2003; Fall et al., 2004). In Webpage classification, SVM has been used on kernels defined on linkage-based similarities and reported good performance (Calado et al., 2006).

In kernel-based methods, we can use well-established kernel composition rules to combine different types of information in a learning task (Cristianini and Shawe-Taylor, 2000; Joachims et al., 2001; Tan and Wang, 2004). In Webpage classification, Joachims et al. adopted such a kernel composition method to consider both direct citation information and content information (Joachims et al., 2001).

3.2.2 Kernel-based Methods on Structure Information

Although feature-based methods have been widely used in classification problems, they are often criticized for requiring explicit feature extraction. It is also difficult to define and extract features from instances with complex structures. This may be one reason that citation networks have been used less in patent classification. Kernel-based methods provide an effective alternative to feature extraction for capturing such complex structure information.

In kernel-based methods different kernel functions have been designed to capture structure information (Gartner, 2003). Among these kernels, the convolution kernel (Haussler, 1999) is one of the most widely used. For objects (data instances) containing a set of sub-objects, convolution kernels calculate the similarities between object pairs by conducting pairwise comparisons between the set of sub-objects they contain. As a special case of convolution kernels, graph kernels are designed for data instances whose sub-objects constitute a graph. The similarity between two graphs can be calculated by comparing the sub-structures in the graphs, such as nodes, paths, and sub-graphs. By representing graphs as random walk paths and conducting pairwise comparison of (matching) random walk paths, graph kernels have been successfully used to classify proteins according to their molecular (graph) structures (Kashima et al., 2003; Le et al., 2004; Borgwardt et al., 2005).

Although previous studies showed the effectiveness of capturing structural information using graph kernels, most of these studies focus on the structure information of the sub-objects contained in data instances. In the patent classification problem, patent citation networks represent the structural information outside of data instances, i.e., the evolution processes of innovations. Few previous studies have made the effort to capture such context structure information for classification purposes.

3.2.3 Research Gaps and Research Questions

Table 3.1 summarizes previous patent classification studies in the context of linked document classification. As an important knowledge management task, patent classification has been studied by a number of researchers. However, most previous studies isolated the knowledge contained in an innovation (patent) from its evolution process and employed only individual patent contents to address the classification problem. Even in the broader literature of linked document classification, use of the knowledge evolution process was limited to direct citations (one-step knowledge transfer). The structure of citation (linkage) networks has not been widely utilized.

Aiming to capture the structure of patent citation networks to alleviate the patent classification problem, I focus on the following two research questions in this research:

Q1. Exploiting the evolution process: Can the methods using citation networks outperform those using only direct citations? Will the features in the directly and indirectly cited patents be helpful for classifying the citing patent?

Q2. Combining an innovation's intrinsic information with its evolution process: Will combining citation information with patent contents improve patent classification performance compared with using citation or content information alone?

Table 3.1 A Summary of Studies on Linked Document Classification

Studies	Domains			Features		Techniques	
	Webpage	Literature	Patent	Document	Citation*	Feature-based	Kernel-based
(Larkey, 1999)			√	√		√	
(Hull et al., 2001)			√	√		√	
(Teichert and Mittermayer, 2002)			√	√		√	
(Krier and Zacca, 2002)			√	√		√	
(Koster et al., 2001, 2003)			√	√		√	
(Fall et al., 2003)			√	√		√	√
(Fall et al., 2004)			√	√		√	√
(Richter and MacFarlane, 2005)			√	√		√	
(Loh et al., 2006)			√	√		√	√
(Chakrabarti et al., 1998)	√		√	√	Neighbor	√	
(Oh et al., 2000)	√			√	Neighbor	√	
(Slattery and Craven, 1998)	√			√	Linkage	√	
(Craven et al., 2000)	√			√	Linkage	√	
(Craven and Slattery, 2001)	√			√	Linkage	√	
(Ghani et al., 2001)	√			√	Both	√	
(Yang et al., 2002)	√			√	Both	√	
(Furnkranz, 2002)	√			√	Neighbor	√	
(Taskar et al., 2002)	√			√	Both	√	
(Joachims et al., 2001)	√			√	Linkage		√
(Calado et al., 2006)	√			√	Linkage	√	√
(Ghanem et al., 2002)		√		√			√
(Sinclair and Webber, 2004)		√		√		√	

* Neighbor: to use neighbors' features; Linkage: to use the linkage itself; Both: to use both kind of features.

3.3 Research Design

To capture the structure of citation networks, I adopt a kernel-based approach, which also enables us to combine citation information with content information.

3.3.1 A Framework of Kernel-based Patent Classification

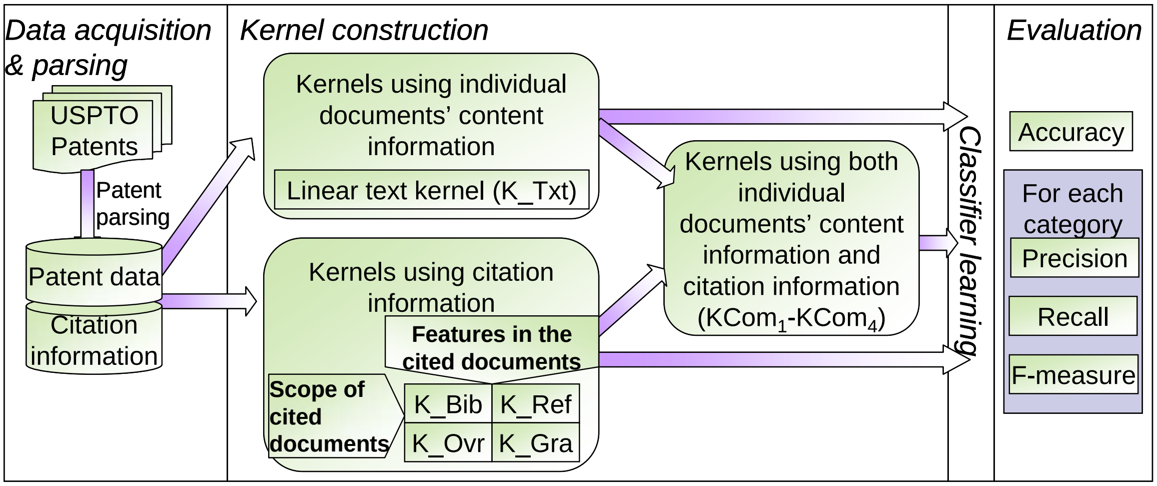

Data acquisition & parsing
USPTO Patents
Patent parsing
Patent data
Citation information
Kernel construction
Kernels using individual documents' content information
Linear text kernel (K_Txt)
Kernels using citation information
Features in the cited documents
Scope of cited documents
K_Bib
K_Ref
K_Ovr
K_Gra
Kernels using both individual documents' content information and citation information (KCom1-KCom4)
Classifier learning
Evaluation
Accuracy
For each category
Precision
Recall
F-measure

Figure 3.1 A Framework of Kernel-based Patent Classification

Figure 3.1 presents a general framework for addressing the patent classification problem using a kernel-based approach. 1) At the data acquisition and parsing stage, patent data are retrieved and parsed into structured data. It should be noticed that both the patents of interest and their directly or indirectly cited patents need to be extracted. 2) At the kernel construction stage, the similarities between data instance pairs are pre-computed according to the kernel function designs. Different kernel functions can capture different information in patents and patent citation networks. 3) At the classifier learning stage, classifiers are learned based on the pre-computed kernel values using a kernel machine. In this research, I chose SVM as the kernel machine because of its reported good performance (Joachims et al., 2001; Fall et al., 2003; Fall et al., 2004; Loh et al., 2006). 4) At the evaluation stage, testing data instances are provide to the classifiers for predictions. The classification performances of different classifiers are evaluated by comparing the predictions against the actual categories provide by experienced patent examiners.

In the proposed kernel-based framework, kernel functions define similarity measures between data instances and capture patterns in data instances. The kernel machine is in charge of building the classification models. The performance of kernel-based methods is highly dependent on the design of kernel functions (Tan and Wang, 2004). The major problem (and contribution) of this research becomes designing appropriate kernel functions for patent classification.

3.3.2 Kernel Function Design

In light of the research gaps, I adopt and design several citation-related kernels that utilize patent citation and content information. Among these kernels, the labeled citation graph kernel is a novel kernel that captures more comprehensive information from the patent citation networks.

3.3.2.1 Using Citation Information

I consider two conditions in the design of citation-related kernels:

1) The scope of the cited documents: The different levels of citations represent the different steps of knowledge transfer. In addition to considering direct citations as an approximation for one-step knowledge transfer, we can extend the citation structure and consider multiple levels of cited documents, which represent a more complete picture of the knowledge evolution process.

2) The features of the cited documents: When modeling an innovation's evolution process, we can choose to use or not use the cited documents' features. Without considering features of cited documents, a patent's cited patents are encoded only as identifiers. If cited documents' features are considered, the semantics of knowledge elements in cited patents are used, which provide extra clues for understanding the focal innovation. In this study I consider the known classification categories of the cited patents as this type of feature, due to reported effectiveness in patent classification (Chakrabarti et al., 1998).

By combining these two conditions, I construct four kernels on patent citation information (see Table 3.2): bibliographic coupling kernel (K_Bib), labeled co-reference kernel (K_Ref), graph overlap kernel (K_Ovr), and labeled citation graph kernel (K_Gra).

Table 3.2 Kernels for Citation Information

	No cited documents' features	Using cited documents' features
Direct citations	Bibliographic coupling kernel (K_Bib)	Labeled co-reference kernel (K_Ref)
Citation network	Graph overlap kernel (K_Ovr)	Labeled citation graph kernel (K_Gra)

a) Bibliographic coupling kernel:

The bibliographic coupling kernel (K_Bib) design adopted from (Calado et al., 2006) was initially used in the context of Webpage classification. It utilizes direct citations of patent documents without considering the cited documents' features. In this kernel, a patent p is represented by a set of patents it cites: $CV_p = \{q : p\ cites\ q\}$. The similarity between two patents is defined as the number of their common citations divided by the total number of their citations:

$$K\_Bib(p_1, p_2) = \left|CV_{p_1} \cap CV_{p_2}\right| / \left|CV_{p_1} \cup CV_{p_2}\right|$$

where p_1 and p_2 are two patents and CV_{p_1} and CV_{p_2} represent the two sets of patents they directly cited. In this kernel, the more common neighbors that two patents share, the more similar they are.

b) Labeled co-reference kernel:

I design a labeled co-reference kernel (K_Ref) to consider cited patents' features (classification category) while using only the direct citations. In this kernel, a patent *p* is represented as a classification category vector, $CC_p = (c_1, c_2, ..., c_n)$, where the elements are the numbers of directly cited patents of *p* that belong to each classification category. The labeled co-reference kernel is defined as the normalized inner product of the classification category vectors:

$$K\_Ref(p_1, p_2) = \left\langle CC_{p_1}, CC_{p_2} \right\rangle / \sqrt{\left\langle CC_{p_1}, CC_{p_1} \right\rangle \bullet \left\langle CC_{p_2}, CC_{p_2} \right\rangle}$$

where p_1 and p_2 are two patents and CC_{p_1} and CC_{p_1} represent their classification category vectors. In the labeled co-reference kernel, if two patents have similar citation patterns in different categories, they have relatively high similarity.

c) Graph overlap kernel:

Based on the bibliographic coupling kernel, I design a graph overlap kernel (K_Ovr) which considers more than one level of the cited patents in the patent citation network. In this kernel, a patent *p* is represented by the set of patents it directly or indirectly cited: $GV_p = \{CV_p \subseteq GV_p;\ if\ \ s \in GV_p\ \ and\ \ s\ \ cites\ \ t\ \ then\ \ t \in GV_p\}$. Two patents' similarity is defined by the ratio of the overlap part of the two patent citation networks in the union of the two networks:

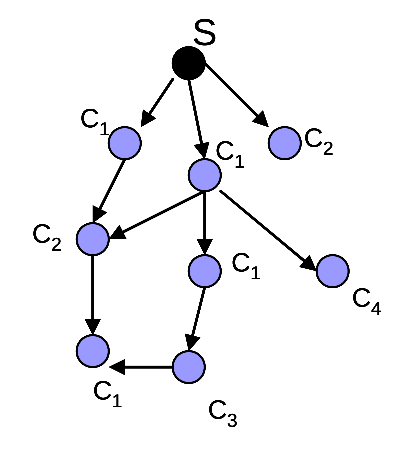
$$K\_Ovr(p_1, p_2) = \left|GV_{p_1} \cap GV_{p_2}\right| / \left|GV_{p_1} \cup GV_{p_2}\right|$$

where $\left|GV_{p_1} \cap GV_{p_2}\right|$ is the number of common patents in the two citation networks, and $\left|GV_{p_1} \cup GV_{p_2}\right|$ is the total number of patents in the two networks. In the graph overlap kernel, the larger the overlap part of the two citation networks, the more similar the two patents are.

d) Labeled citation graph kernel:

Lastly, as my main contribution, I propose a labeled citation graph kernel (K_Gra) which considers both the network of cited documents and the cited documents' features. In this kernel, a patent *p* is associated with a labeled citation network, $G_p := (GV_p, GE_p, GL_p)$, which contains the patents directly or indirectly cited by *p*: GV_p, and the citations between all patents in $GV_p : GE_p = \{(s,t) : \forall s,t \in GV_p \text{ and } s \textit{ cites } t\}$. In this network, each node (patent) is labeled with its classification category: $GL_p = \{label(q) : \forall q \in GV_p\}$. The similarity between two patents is measured by the similarity between the labeled citation networks associated with them.

In order to analyze patents using their associated labeled citation networks, the labeled citation graph kernel compares the random walk paths, starting from the focal patents on their associated labeled citation networks, and composes path similarities into the similarities of focal patents. This is different from previous graph kernel studies that target analyzing graphs, which compare random walk paths starting from any nodes in the focal graphs (Kashima et al., 2003). The random walk paths are generated from the focal patents following patent citations (Figure 3.2). When a random walk is conducted, it follows a probability distribution and may jump from one patent to one of its neighbors (cited documents) or stop at the patent. From a knowledge diffusion perspective, the random walk paths represent the knowledge transfer paths (reversely) from prior innovations to focal patents. In this model, a longer random walk path has a lower probability of existence, indicating the less impact of older predecessors on new innovations.



Random walk paths
1. S→C_1
2. S→C_1
3. S→C_2
4. S→C_1→C_2
5. S→C_1→C_2
6. S→C_1→C_1
7. S→C_1→C_4
8. S→C_1→C_2→C_1
9. S→C_1→C_2→C_1
10. S→C_1→C_1→C_3
……

Figure 3.2 Random Walk Paths on a Labeled Citation Network Related to Patent S

In the labeled graph kernel, each random walk path is represented as a sequence of labels (i.e., classification categories) of the nodes on the paths, which partially documents the knowledge elements related to this knowledge transfer path. The similarity of two paths is considered to be one if they share identical label sequences. Otherwise, it is considered to be zero for the sake of simplicity. The labeled citation graph kernel is defined as the sum of pairwise path similarity values, which are weighted according to the probabilities these random walk paths may exist. In other words, the kernel compares all knowledge transfer paths leading to each innovation to identify patents on similar topics. The algorithm to calculate the labeled citation graph kernel is summarized in Figure 3.3.

1. Random path generation

   1) The random walk starts from the patent to be classified x_0.

   2) On node x_i the random walk has a probability of $p_q(x_i)$ to stop.

   3) If the random walk does not stop, the random walk has equal probability to choose any of x_i's neighbors (which is noted as x_{i+1}) to jump to. The probability is noted as $p_t(x_{i+1}|x_i)$.

   4) Thus a random walk path $h=\{x_0, x_1, \ldots, x_n\}$ has the probability $P(h|G)=p_t(x_1|x_0)p_t(x_2|x_1)\ldots p_t(x_n|x_{n-1})p_q(x_n)$ to exist.

2. Kernel definition

   The labeled citation graph kernel is defined as a convolution kernel

   $$K\_Gra(G_{p_1}, G_{p_2}) = \sum_h \sum_{h'} k(h,h')P(h \mid G_{p_1})P(h' \mid G_{p_2})$$

   For two random walk paths $h=\{x_0, x_1, \ldots, x_n\}$ and $h'=\{x'_0, x'_1, \ldots, x'_n\}$

   if n<>m, $k(h,h')$=0,

   else $k(h,h') = \prod_{i=1}^{n} \hat{k}(x_i, x_i')$ where $\hat{k}(x_i, x_i') = 1$ *iff* label(x_i)=label(x'_i).

Figure 3.3 Labeled Citation Graph Kernel Algorithm

3.3.2.2 Using Individual Documents' Content Information

The kernel that uses individual documents' content information represents the previous efforts that used content features to address the patent classification problem. In previous studies, features extracted from patent abstracts, claims, and descriptions have all been used. Patent abstracts have been reported to be slightly more informative than other features in patent classification (Larkey, 1999; Loh et al., 2006). The linear text kernel has been reported to have good classification performance (Fall et al., 2003; Fall et al., 2004). Therefore, I use the patent abstract to represent the entire patent content and choose the linear text kernel to capture patent content information. Such a setting works as a baseline to evaluate the performances of the citation-based kernels. In the linear text kernel, each patent p is represented by a term vector, $C_p = (t_1, t_2, ..., t_m)$, where the elements are the number of occurrences of terms in the abstract. The linear text kernel (K_Txt) defines the similarity of two patents as the normalized inner product of the term vectors (Joachims et al., 2001):

$$K\_Txt(p_1, p_2) = \left\langle C_{p_1}, C_{p_2} \right\rangle / \sqrt{\left\langle C_{p_1}, C_{p_1} \right\rangle \bullet \left\langle C_{p_2}, C_{p_2} \right\rangle}$$

where p_1 and p_2 represent two patents and C_{p_1} and C_{p_2} are their term vectors.

3.3.2.3 Using Both Content & Citation Information

Using kernel composition methods, it is easy to consolidate different types of information by combining multiple kernels. I use the simple addition operation to combine kernels that use citation information (K_Bib, K_Ref, K_Ovr and K_Gra) with the linear text kernel (K_Txt) into four composite kernels (K_Com_1-K_Com_4). For any two kernel functions *K₁(p1, p2)* and *K₂(p1, p2)*, the addition operation creates a new kernel: $K(p_1, p_2) = \lambda K_1(p_1, p_2) + (1-\lambda) K_2(p_1, p_2)$. The addition operation on the two kernels implicitly combines the feature spaces defined by them. The parameter λ controls how much each kernel contributes to the composite kernel. This set of kernels represents the efforts that exploit both patent contents and the associated knowledge evolution process.

3.4. Experimental Study

3.4.1 Dataset

In order to examine the effectiveness of proposed kernel functions for patent classification, I conducted an experimental study on a nanotechnology-related patent dataset acquired from the USPTO. I chose USPTO patents because they have more complete citation information than patents from other patent offices (hence more reliable citation networks). I selected patents in a specific domain so as to restrict the size of the testbed without significantly reducing the difficulty of the patent classification task. Specifically, nanotechnology was selected due to its deep impact on a nation's technology advancement and its rapid development in patent publication in recent years, reflecting the characteristics of many high-tech domains.

I retrieved nanotechnology-related patents from the USPTO by keyword-searching in patent title, abstract, and claims, using a keyword list provided by domain experts (Huang et al., 2003b). The retrieved patents were parsed into structured data and stored in a relational database. I also retrieved the patents they directly or indirectly cited to reconstruct the citation network. Since the number of cited patents increases exponentially as the citation level increases, from a practical standpoint I retrieved only cited patents that are two steps away from the core set of patents. (The testbed may contain a patent's ancestors that are more than two steps away, if it cites the patents in the core set of patents.)

Table 3.3 Number of Data Instances in the Testing and Training Datasets

	Number of patents	Number of categories	Number of patents in the citation network	Number of categories in the citation network
Training	13,913	36	336,303	426
Testing	4,358	36	227,833	410

I split the testbed into a training set and a testing set following previous studies (Krier and Zacca, 2002). Given a specific date, patents published prior to that date were used as the training data, while applications filed after that date were used as testing data. The patents under review on that day, which have been applied for but have not yet been issued, were not considered in either the training or testing dataset. In this research, the patents published between 01/01/1999 and 12/31/2001 were used for training. The patent applications that were filed between 01/01/2002 and 12/31/2004 were used for testing. I used a patent's major USPC category as its classification label. To provide enough instances to train the classifier, I restricted the experiments to categories with more than 100 patents in the training dataset. After preprocessing, the training dataset contained 13,913 data instances and the testing dataset contained 4,358 data instances (see Table 3.3) which belong to 36 first-level USPTO categories. The number of instances in each category varied from 109 to 1,895 in the training data and from 15 to 705 in the testing data (Figure 3.4). The retrieved citation network of the training set contained 336,303 patents, and that of the testing set contained 227,833 patents. As there were overlap patents in these two citation networks, in total I collected 451,853 patents.

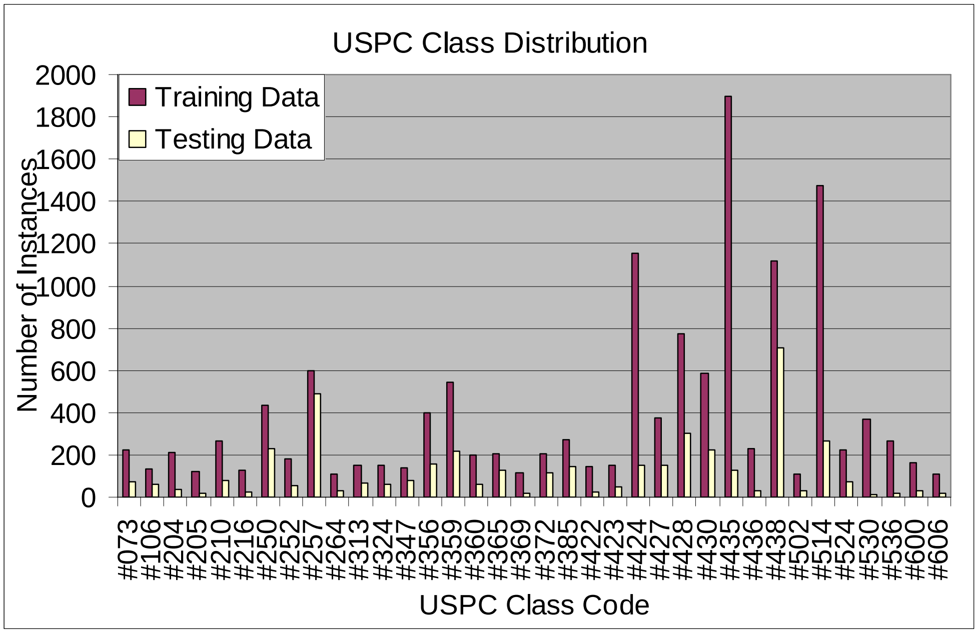


Figure 3.4 Data Distribution in USPC Categories

The research testbed illustrates the challenges in patent classification discussed earlier. Produced by a multi-disciplinary research field, nanotechnology patents cover many USPC categories (Huang et al., 2003b). Some of these patents may have minor topical differences and are difficult to differentiate. In the dataset, the numbers of instances are uneven in different categories. This dataset contains 36 classification categories, which is comparable to previous patent classification studies.

3.4.2 Experimental Procedures

After creating the training and testing datasets, I calculated the kernel matrices that contain the kernel values between the patents in the datasets. To construct the linear text kernel matrix, I preprocessed the contents (abstracts) of the patents in the testbed using the open source package “Rainbow” (McCallum, 1996) for stemming, indexing, and feature selection (based on mutual information). To construct the kernel matrices of citation information, I used the extracted citation relations and classification categories of cited patents and pre-computed the kernel values according to their definitions. To construct the four composite kernels, I set λ as 0.5 and added the linear text kernel matrix to each of the four kernel matrices of citation information. I choseλ as 0.5 for consistency with past research (Joachims et al., 2001), where individual documents’ content and citation information have equal effect on the final kernel matrix. It is worth knowing that the parameter λ can be optimized by solving a semidefinite programming problem (Lanckriet et al., 2004). However, parameter optimization is out of the scope of the current research and will be considered in the future. After the pre-computation of kernel matrices, I used a well-known high-performance SVM package, “libSVM” (Chang and Lin, 2001), to build the classification models. I classify each patent to only one class, which is considered as its major classification category. The predictions on the testing dataset are used for evaluation.

3.4.3 Evaluation Metrics

For each of the data instances in the testing dataset, I compared the classifiers’ predictions with its actual classification category in the USPTO. I used standard classification performance metrics, accuracy, precision, recall, and F-measure, to evaluate the performance of different kernels with the SVM algorithm. These metrics have been widely used in information retrieval and machine learning studies.

Accuracy is usually used to assess the overall performance of a classifier at the instance level. For the instances in the testing set,

$$\text{Accuracy} = \frac{\text{number of all correctly identified instances}}{\text{total \# of instances}}.$$

Precision, recall, and F-measure are defined to evaluate the performance of a classifier on individual classes. For a class *i*, if TP_i is the number of correctly identified instances of class *i*, FP_i is the number of instances incorrectly assigned to class *i*, and FN_i is the number of instances which belong to class *i* and have been assigned to other classes by mistake, then

Precision P_i= $TP_i / (TP_i + FP_i)$,

Recall R_i= $TP_i / (TP_i + FN_i)$, and

F-measure F_i= $2 \times P_i \times R_i / (P_i + R_i)$, which combines precision and recall.

The micro-average (per-instance) value and macro-average (per-category) value of precision, recall, and F-measure can be used to compare the kernels' overall performances (Yang, 1999; Sebastiani, 2002). Given that the experiments are designed as single-label classification, the micro-averaged precision, recall, and F-measure are equal to accuracy, which favors the categories with large numbers of instances by giving each instance the same weight. Thus, I report the macro-averaged precision, recall, and F-measure, which favor the categories with small numbers of instances since each category has the same weight.

3.4.4 Hypotheses

In correspondence with the research questions, I tested two sets of hypotheses to examine the effects of using citation networks in patent classification. In these hypotheses, I adopted (a) accuracy and (b) F-measure (which combines the precision and recall) to gauge the instance-level and category-level performances of different settings.

H1.1a. Kernels that use the structures of patent citation networks will outperform those that use only direct citations on classification accuracy in patent classification.

H1.1b. Kernels that use the structures of patent citation networks will outperform those that use only direct citations on F-measure in patent classification.

H1.2a. Kernels that use classification categories as cited documents' features will outperform those that do not use any cited documents' features on classification accuracy in patent classification.

H1.2b. Kernels that use classification categories as cited documents' features will outperform those that do not use any cited documents' features on F-measure in patent classification.

H2.1a. Composite kernels of citation information and patent content will outperform the linear text kernel that uses patent contents on classification accuracy in patent classification.

H2.1b. Composite kernels of citation information and patent content will outperform the linear text kernel on patent contents on F-measure in patent classification.

H2.2a. Composite kernels of citation information and patent content will outperform kernels that use only citation information on classification accuracy in patent classification.

H2.2b. Composite kernels of citation information and patent content will outperform kernels that use only citation information on F-measure in patent classification.

I conducted single-sided pairwise t-tests to test these hypotheses. The t-test on accuracy was conducted at the instance level, in which the mean of every instance's correctness (0 or 1) is accuracy. The t-test on F-measure was conducted at the category level, in which the mean of every class's F-measure is the macro-averaged F-measure.

3.5. Results and Discussion

3.5.1 Overall Performances

Table 3.4 Performances of Different Kernels

Kernels	Accuracy	Averaged precision	Averaged recall	Averaged F-measure
Bibliographic coupling kernel (K_Bib)	7.48%	47.87%	5.81%	5.71%
Labeled co-reference kernel (K_Ref)	61.50%	56.04%	56.82%	55.50%
Graph overlap kernel (K_Ovr)	37.13%	53.32%	29.08%	34.91%
Labeled citation graph kernel (K_Gra)	86.67%	89.09%	87.97%	88.04%
Composite kernel 1 (K_Com$_{1}$)	57.82%	53.65%	44.24%	46.50%
Composite kernel 2 (K_Com$_{2}$)	66.02%	59.43%	59.14%	58.78%
Composite kernel 3 (K_Com$_{3}$)	59.64%	55.49%	47.56%	49.72%
Composite kernel 4 (K_Com$_{4}$)	87.84%	89.43%	86.97%	87.96%
Linear Text Kernel (K_Txt)	55.55%	51.65%	39.29%	40.83%

Table 3.4 reports the performances achieved by the SVM classifiers with different kernels. We can observe that both the labeled citation graph kernel (K_Gra) and its composition with the linear text kernel (K_Com$_{4}$) have high accuracies, precisions, recalls, and F-measures. They achieve much better performances (31.12% and 32.29%

absolute improvement in accuracy) than the baseline linear text kernel (K_Txt). Considering that the linear text kernel represents the performance of content analysis (using the knowledge embedded in patents) in previous research and applications, the two kernels show good potential to be used in real applications. Both kernels utilize the network structure of patent citations and the classification category features of cited documents, which account for their good performances.

In the experiments, the bibliographic coupling kernel (K_Bib) and the graph overlap kernel (K_Ovr) have low accuracy values (7.48% and 37.13%, respectively). This may be a direct result of their sparse kernel matrices. The designs of these two kernels compare patent citations according to exact match. Given the millions of patents existing in the world, the probability that two patents share the same references is very low. Thus, there is a high probability that the kernel values will be zero. In the experiments, the bibliographic coupling kernel has 99.88% zero values and the graph overlap kernel has 98.37% zero values. Compared with the linear text kernel whose matrix has 38.81% zero values, the two kernel matrices are too sparse to capture enough information to differentiate patents and build an effective classifier.

It is also noticed that the bibliographic coupling kernel (K_Bib) and the graph overlap kernel (K_Ovr) have much lower recalls (5.81% and 29.08%) than the other kernels, while most kernels have similar precision values (except the labeled citation graph kernel and its composition with the linear text kernel). Further inspection shows that the two kernels tend to assign most patents into certain classes by mistake. For example, the bibliographic coupling kernel assigns most instances into USPC category

#435 (Chemistry: molecular biology and microbiology) with a low precision. The few instances left were assigned accurately, which lead to a high precision and a very low recall in most classes. For example, the bibliographic coupling kernel has 100% precision in assigning a couple of instances into some categories (e.g., 3 instances in USPC category #073, 3 instances in USPC category #106, and 1 instance in USPC category #252).

3.5.2 Hypotheses Testing

To further assess the factors that affect the performances of different kernels, I tested the hypotheses by conducting single-sided pairwise t-tests on accuracy and F-measure (Table 3.5). The pairwise t-tests on accuracy were conducted at the instance level (n=4,358); the pairwise t-tests on F-measure were conducted at the class level (n=36).

Statistical tests confirm that the kernels that use networks of patent citations significantly outperform the kernels that use only direct citations on both accuracy and F-measure (i.e., H1.1a and H1.1b are supported). Using citation networks explicates the relationship between the patents which do not share directly cited patents but share indirect ancestors. Such explications may provide more evidence when the classifiers try to categorize such patents into the same class. In addition, using citation networks differentiates the patents with similar directly cited patents more distinctly by inspecting more levels of citations. Such detailed differentiation may enable the classifiers to categorize ambiguous patents into different classes more precisely.

Table 3.5 Hypotheses Testing for Different Kernels

H1.1: p values	a) Pairwise t-test on accuracy	b) Pairwise t-test on F-measure
K_Bib < K_Ovr	<0.001	<0.001
K_Ref < K_Gra	<0.001	<0.001

H1.2: p values	a) Pairwise t-test on accuracy	b) Pairwise t-test on F-measure
K_Bib < K_Ref	<0.001	<0.001
K_Ovr < K_Gra	<0.001	<0.001

H2.1: p values	a) Pairwise t-test on accuracy	b) Pairwise t-test on F-measure
K_Txt < K_Com$_1$	<0.001	<0.001
K_Txt < K_Com$_2$	<0.001	<0.001
K_Txt < K_Com$_3$	<0.001	<0.001
K_Txt < K_Com$_4$	<0.001	<0.001

H2.2: p values	a) Pairwise t-test on accuracy	b) Pairwise t-test on F-measure
K_Bib < K_Com$_1$	<0.001	<0.001
K_Ref < K_Com$_2$	<0.001	<0.005
K_Ovr < K_Com$_3$	<0.001	<0.001
K_Gra < K_Com$_4$	0.004	0.533

Statistical tests confirm that the kernels that use cited documents' classification category features significantly outperform those that do not use any cited documents' features on both accuracy and F-measure (i.e., H1.2a and H1.2b are supported). Previous research found that employing neighbor documents' classification category information can improve the classification accuracy (Chakrabarti et al., 1998; Oh et al., 2000), which is confirmed by my experiments. My experiments further suggest that, when the entire citation network is considered, cited documents' features can still play an important role.

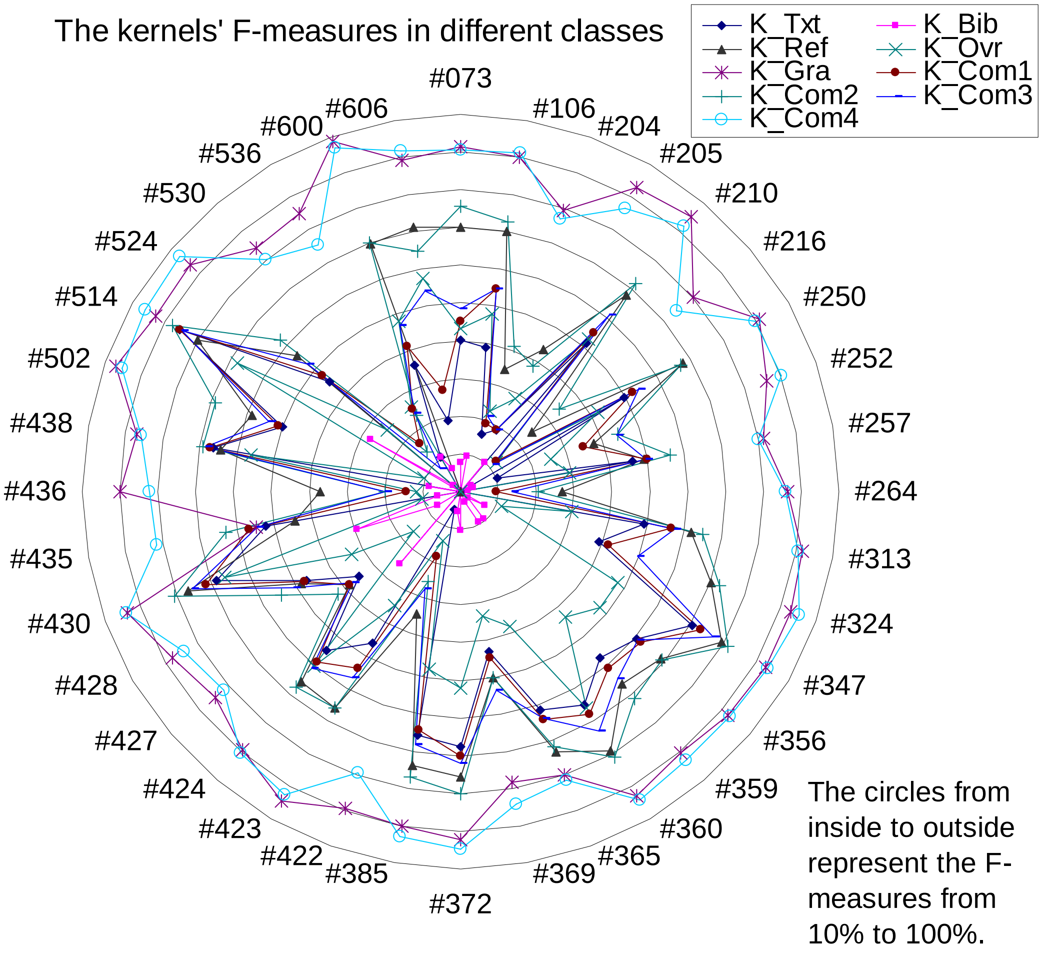
Statistical tests show that all four composite kernels significantly outperform the linear text kernel (K_Txt) on both classification accuracy and F-measure (i.e., H2.1a and H2.1b are supported). In the statistical test to compare composite kernels with the kernels using only citation information, although the labeled citation graph kernel and its composition with the linear text kernel do not have statistically significant differences in F-measures in the testing of H2.2b (p value ≈ 0.533), all other tests on accuracy and F-measure confirm a better performance when combining information (i.e., H2.2a is supported and H2.2b is partially supported). The statistical test results strongly suggest the complementary roles of patent citations and patent contents when used in patent classification tasks. In the experiments, the bibliographic coupling kernel (K_Bib) and the graph overlap kernel (K_Ovr) achieved only 7.48% and 37.13% accuracy, respectively. However, when they were combined with the linear text kernel, the classification performance improved significantly. This indicates that even though the citation information may be sparse in patents and using it alone is not very helpful, combining citation and content information can still improve the performance for patent classification.

3.5.3 Individual Class's Performances

Table 3.6 Some of the Categories Which are Difficult to Classify

USPC code	Category description	Number of training instances	Number of testing instances
#216	Etching a substrate: processes	124	23
#264	Plastic and nonmetallic article shaping or treating: processes	111	33
#422	Chemical apparatus and process disinfecting, deodorizing, preserving, or sterilizing	143	23
#436	Chemistry: analytical and immunological testing	229	28
#530	Chemistry: natural resins or derivatives; peptides or proteins; lignins or reaction products thereof	367	15
#536	Organic compounds -- part of the class 532-570 series	265	18

I also inspected the kernels' performances on all 36 classes. Figure 3.5 shows the F-measure each kernel achieved in each class. In general, the labeled citation graph kernel (K_Gra) and its composition with the linear text kernel (K_Com4) have high performance in most of the 36 categories. However, the F-measures of the other seven kernels vary in the 36 categories, which may reduce their usability. The seven kernels' F-measures are relatively low in a similar group of categories. Table 3.6 provides some examples of these categories, which are difficult to classify and have a relatively small number of training instances. The testbed includes other categories which share similar topics with these categories and have a larger number of training instances. The classifiers have a high probability of misclassifying patents belonging to these categories into other similar categories. For example, most of the instances in USPC category #216 (Etching a substrate: processes) were incorrectly assigned to category #438 (Semiconductor device manufacturing: process), which has 1,119 training instances. Many of the instances in category #264 (Plastic and nonmetallic article shaping or treating: processes) were assigned to category #428 (Stock material or miscellaneous articles), which has 774 training instances. Many of the instances in categories #422 (Chemical apparatus and process disinfecting, deodorizing, preserving, or sterilizing), #436 (Chemistry: analytical and immunological testing), #530 (Chemistry: natural resins or derivatives; peptides or proteins; lignins or reaction products thereof), and #536 (Organic compounds -- part of the class 532-570 series) were assigned to #435 (Chemistry: molecular biology and microbiology), which has 1,895 training instances. Even in these categories where most other kernels fail, the labeled citation graph kernel and its composition with the linear text kernel (K_Com4) are highly accurate. By considering patent citation information, the two kernels have better differentiation abilities on the categories with very similar topics and uneven numbers of instances.



The circles from inside to outside represent the F-measures from 10% to 100%.

Figure 3.5 The Kernels' Performances in Different Classes

The performance of the labeled citation graph kernel (K_Gra) and its composition with the linear text kernel (K_Com4) also changes slightly in different categories. In Figure 3.5, the labeled citation graph kernel (K_Gra) does not achieve a high performance in USPC category #435 (F-measure=54.71%). Although it is better than most of the other kernels in the same category, such a performance is not comparable to the performance it achieved in other categories (F-measures between 77.78% and 98.31%). USPC category #435 has the largest number of training instances in the dataset and a small number of testing instances. The patents in this category are on fundamental science topics or research tools, which were heavily cited by patents in all categories (Li et al., 2007). These characteristics may be the cause of the low performance of the labeled citation graph kernel and other kernels in USPC category #435. However, after combining it with the linear text kernel, the composite kernel (K_Com4) achieves a high F-measure on USPC category #435 (81.25%). The composite kernel (K_Com4) employs content features in addition to citations, which may help the classifiers differentiate the patents belonging to USPC category #435 from the others. Actually, the composite kernel (K_Com4) achieves consistent good performance in all categories (F-measures between 74.42% and 96.73%, average F-measure 87.96%, standard deviation 6.53%). Even the labeled citation graph kernel is highly accurate; considering patent contents (linear text kernel) ensures more consistent high performance for different categories.

3.6 Summary

Using patent classification as an example, this chapter demonstrates that knowledge evolution processes embedded in patent citation networks can be modeled and utilized in knowledge management tasks. In this research, I designed different kernel functions under a kernel-based framework to capture citation network information for patent classification. The proposed labeled citation graph kernel significantly improved patent classification performance. The research shows that the features of cited patents and the structure of patent citation networks, which together represent innovations' evolution history, can benefit the classification of focal patents. It is also noticed that combining the information in citation networks with patent contents results in higher and more consistent performance.

In the practice of patent management, the significant performance improvement (>30% in accuracy) in the experiments indicates the good potential of using the proposed approach to alleviate human efforts in patent pre-classification and further expedite patent examination. The research also lends support to a policy that requires inventors to file patent citations, since they often have more complete knowledge about their innovation's evolution.

The proposed approach can be directly applied to classify other linked documents, such as Webpages and scientific literature. With appropriate adaptations it is also applicable to other knowledge codification and organization tasks such as building help desk systems, decision support systems, and knowledge repositories.

The effectiveness of the proposed approach shows the importance of considering the network structure in the node classification task. The proposed model explores features from related nodes to build models for focal nodes' analysis. In patent classification, such a model can be interpreted with knowledge diffusion theory. In the next chapter, I will show that the use of network structure is not limited to such theories. I will examine the use of related nodes' features from a context perspective and conduct the analysis in a gene function prediction application.

CHAPTER 4. NODE CLASSIFICATION: CAPTURING GENE INTERACTION CONTEXTS FOR GENE FUNCTION PREDICTION

4.1 Introduction

In the previous chapter, I proposed a graph-based model based on the knowledge creation and diffusion theory for node classification. In this chapter, I generalize this model and propose to capture nodes' contexts for classification purposes. The generalized model (context graph kernel) can be used in a broad range of applications, such as the gene function prediction problem in this chapter. I also deduce the matrix formulation of this model and analyze the characteristics of the model under different parameter settings.

In recent years, developments in genome sequencing have led to the identification of a large number of genes. However, most of these genes' functions remain poorly known or unknown (Enright et al., 2003). Annotating genes' functions has become a major challenge for biologists in the post-genomic era, which need more development on computational techniques. In the early stages of computational modeling, individual genes' physical, chemical, and biological characteristics were the major features used for function prediction. Recent studies have used gene interaction information and obtained promising results (Hu et al., 2007). However, in most of these studies gene interactions are considered to be indicators of functional similarities between connected genes, which restrict the prediction power of the models.

In this chapter, I propose to predict a gene's functions according to its context graph, which is defined as the gene interaction network composed of the genes interacting directly and indirectly with the focal gene. I propose a context graph kernel in a kernel-based machine learning framework that uses both gene features (node information) and structural characteristics of the context graph to infer the focal gene's functions.

This chapter is organized as follows. Section 4.2 reviews related studies on gene function prediction using gene interaction information. Section 4.3 introduces the proposed context graph kernel method. Section 4.4 describes the experiments on a p53-related dataset. Section 4.5 discusses the results. Section 4.6 summarizes the findings.

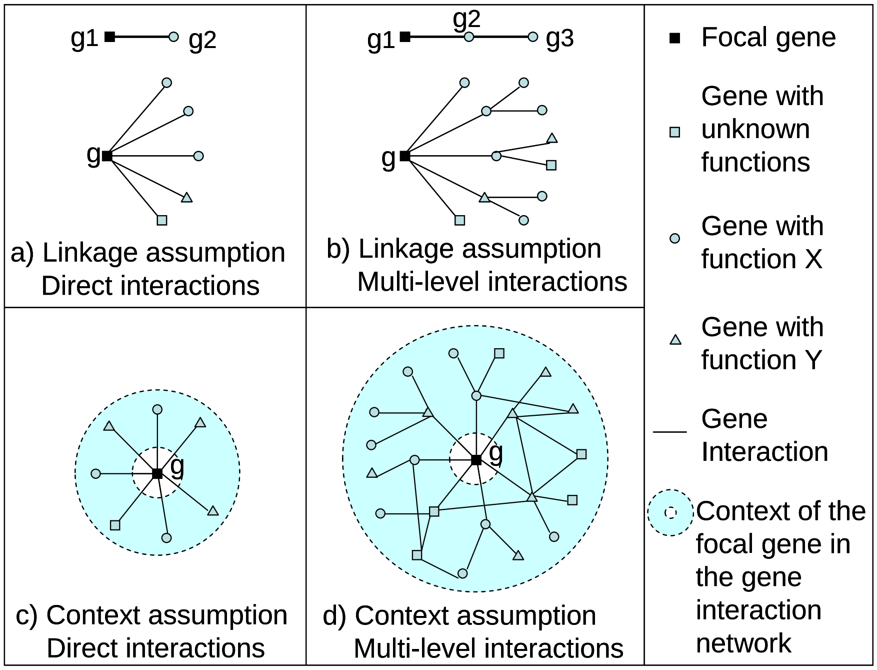
4.2 Literature Review

4.2.1 Gene Function Prediction

Gene functions can be predicted through annotating individual genes (Sharan et al., 2007) or gene clusters (D'haeseleer et al., 2000; Huynen et al., 2003). At the individual gene level, gene features such as gene sequence (Altschul et al., 1997; Jensen et al., 2003), molecule structure (Borgwardt et al., 2005), and gene co-expression patterns (Pavlidis et al., 2002) have been used for annotation. Recent studies observed that gene interactions in biological pathways (including gene-gene interactions, gene-protein interactions, and protein-protein interactions) are also related to the functions of genes. Thus, the significant amount of gene interactions (Karaoz et al., 2004; Li et al., 2006) found in previous biology research can become another important resource for gene function prediction.

I review previous interaction-based function prediction studies along three dimensions: assumptions, levels of interactions, and computational techniques.

4.2.1.1 Assumptions



a) A linkage assumption assumes connected genes have similar functions. b) Indirect neighbors may have lower probability of sharing similar functions. c) A context assumption assumes that a gene's functions are correlated with the patterns of its context. d) When multiple levels of gene interactions are used, genes with similar context graphs may have similar functions.

Figure 4.1 Assumptions of Using Gene Interactions in Gene Function Prediction

Previous studies are typically built upon a linkage assumption or a context assumption. A linkage assumption considers gene interactions as an indicator of a functional similarity between connected genes. This assumption is supported by the fact that immediate neighbor genes (Schwikowski et al., 2000) and level-2 neighbor genes (Chua et al., 2006, 2007b) have a high probability of sharing functions. Based on this assumption, a focal gene's functions can be adopted from the majority of its neighbor's functions (Figure 4.1a). In the case of multi-level interactions (Figure 4.1b), it is typically assumed that indirect neighbors have weaker influence on the focal gene's functions.

The context assumption focuses on the correlation between a focal gene's functions and the pattern of its context, i.e., its direct/indirect neighbors. For example, if the functions of a gene's direct/indirect neighbors follow a certain combination, the gene may be assigned to a function which is not the same as the majority of its neighbors' (Figure 4.1c). The topological patterns of gene interaction networks could also be used if multi-level interactions are considered (Figure 4.1d). In previous research, Schlitt et al. considered using direct neighbors as the context and predicted similar functions for genes with similar neighbors (Schlitt et al., 2003).

4.2.1.2 Levels of Interactions

When using gene interactions for function prediction, both direct interactions between neighbor genes and multiple levels of interactions between indirect neighbor genes have been used in previous research. Early studies based on direct interactions used the "guilt by association" rule under the linkage assumption to infer a focal gene's functions as the most frequent ones among neighbors (Mayer and Hieter, 2000; Li et al., 2007c).

Considering multiple levels of interactions is a natural extension. Under the linkage assumption, the genes that are farther from the focal gene may have less impact on the focal gene's function prediction (Figure 4.1b). Features describing second-level neighbors have been explicitly extracted and used in gene function prediction (Chua et al., 2006; Xu and Li, 2006; Chua et al., 2007b). In addition, Hishigaki et al. proposed searching multiple levels of neighbors for the most frequent functions to predict labels of the focal gene (Hishigaki et al., 2001). Some studies further extend the scope of neighbors to the entire gene interaction network, e.g., by minimizing inconsistent function assignments of connected genes in the network (Vapnik, 1995; Karaoz et al., 2004) or through propagating gene function labels over the network with a damping effect (Nabieva et al., 2005).

4.2.1.3 Computational Techniques

Computational techniques for gene function prediction can be categorized into heuristic approaches and machine learning approaches.

Heuristic approaches usually predefine rules to make predictions. For example, after defining the label propagation rule, function labels can be propagated through direct interactions (Mayer and Hieter, 2000) or the entire interaction network (Nabieva et al., 2005) to the genes with unknown functions. The predefined rules can also be used to design objective functions for optimization models. For example, based on the "guilt by association" rule, function prediction is formulated as minimizing inconsistent function assignments of connected genes (Vazquez et al., 2003; Karaoz et al., 2004; Massjouni et al., 2006; Murali et al., 2006). Simulated annealing (Vazquez et al., 2003) and iterative local search methods (Karaoz et al., 2004; Massjouni et al., 2006; Murali et al., 2006) have been proposed to find solutions for such models.

Machine learning approaches build prediction models from patterns in training instances. In particular, kernel-based methods have been frequently used in gene function prediction, due to their ability to capture structural information. Based on the context assumption, linear kernels (Lanckriet et al., 2004) and graph overlap similarities (Zhao et al., 2008) have been used to model focal genes' contexts for function prediction. Based on the linkage assumption, diffusion kernels have been used to model genes' positional characteristics in gene networks for prediction (Tsuda and Noble, 2004; Yamanishi et al., 2004).

4.2.2 Research Gaps

Table 4.1 A Summary of Previous Studies

Studies	Assumption *	Level of Interactions #	Technique +	Descriptions
(Mayer and Hieter, 2000)	L	D	H	Guilt by association (majority voting)
(Schwikowski et al., 2000)	L	D	H	Guilt by association
(Hishigaki et al., 2001)	L	M	H	Multi-level neighbor majority voting
(Schlitt et al., 2003)	C	D	H	Similarity of neighbors
(Vazquez et al., 2003)	L	M	H	Minimize un-matching gene pairs
(Karaoz et al., 2004)	L	M	H	Minimize un-matching gene pairs
(Lanckriet et al., 2004) (a)	L	M	ML	Diffusion kernel (classification by gene positions in the network)
(Lanckriet et al., 2004) (b)	C	D	ML	Linear kernel
(Tsuda and Noble, 2004)	L	M	ML	Locally constraint diffusion kernel
(Yamanishi et al., 2004)	L	M	ML	Diffusion kernel
(Nabieva et al., 2005)	L	M	H	Label propagation
(Massjouni et al., 2006)	L	M	H	Minimize un-matching gene pairs
(Chua et al., 2006)	L	M	H	Weighted neighbor function label counting
(Murali et al., 2006)	L	M	H	Minimize un-matching gene pairs
(Xu and Li, 2006)	B	M	ML	KNN with neighbor-related features
(Chua et al., 2007b)	L	M	H	Weighted neighbor function label counting
(Li et al., 2007c)	L	D	H	Guilt by association
(Zhao et al., 2008)	C	D	H	Similarity of neighbors

* L – linkage assumption; C – context assumption; B – both assumptions.
# D – direct interactions; M – multi-level interactions.
$^{+}$ H – heuristic approach; ML – machine learning approach.

Table 4.1 summarizes previous studies that use gene interactions in gene function prediction. I identify the following research gaps in previous research:

1) Most studies addressed the gene function prediction problem under a linkage assumption. The use of context assumption is limited.

2) Although both direct interactions and multi-level interactions have been used under the linkage assumption, the context-based studies used only direct interactions. The effect of indirect interactions under the context assumption still remains to be investigated.

3) Heuristic approaches have been the major technique adopted. Several machine learning-based studies used features of individual genes (Pavlidis et al., 2002; Jensen et al., 2003; Borgwardt et al., 2005) without considering gene interactions. However, less attention has been paid to leveraging graph structures in statistical learning for gene function prediction.

4.3 Research Design

To bridge the aforementioned research gaps, this study aims at predicting a gene's functions based on its context in a gene interaction network. I also inspect the effect of using multiple levels of (indirect) interactions in function prediction. I choose a kernel-based machine learning approach in this research due to its documented good performance and ability to handle structural data (Gartner, 2003).

4.3.1 A Kernel-based Approach

I formulate gene function prediction as a classification problem. The objective is to assign function labels to each gene. Figure 4.2 shows the process of gene function prediction in a kernel-based framework. At the data preparation stage, gene interactions are extracted from public databases. Genes are annotated with their known function labels. At the classifier construction stage, genes with known functions are selected and used as training instances to build the models. Specifically, I build a binary classifier for each function label. At the inference stage, the genes with unknown functions are given to the classifiers; the predictions from multiple binary classifiers are combined to assign functions to each gene. Finally, at the evaluation stage, the predictions are validated against existing knowledge, by domain experts or through further experiments.

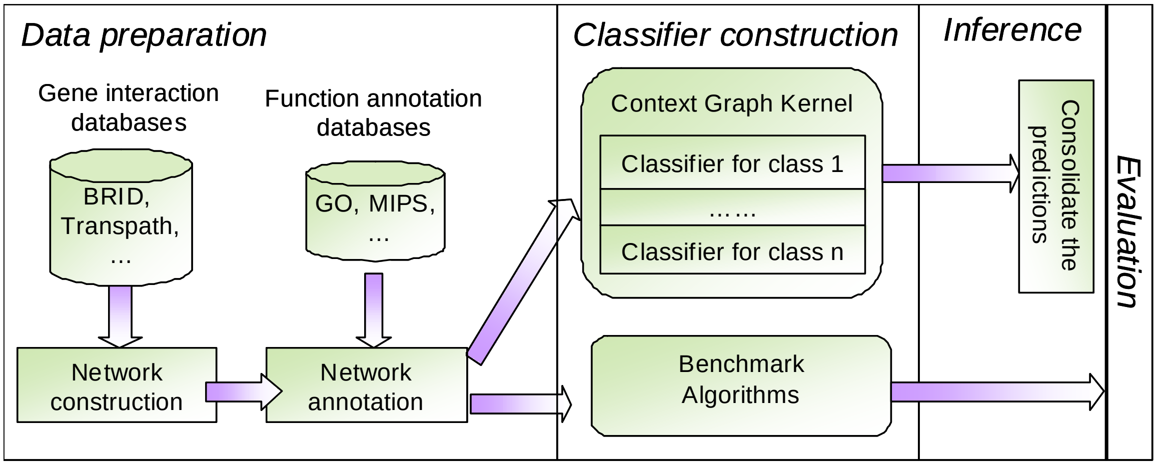


Figure 4.2 The Kernel-based Framework for Gene Function Prediction

To build classifiers using a kernel-based method, a kernel function and a kernel machine need to be specified. The performance of kernel-based methods is highly dependent on the design of kernel functions (Tan and Wang, 2004). Thus, the main focus and contribution of this research is to design a kernel function that can better capture structural patterns in gene interaction networks for the gene function prediction task. For the kernel machine, I choose the well-studied Support Vector Machines (SVM) (Vapnik, 1995) algorithm due to its reported competitive performance in many domains (Muller et al., 2001; Vinayagam et al., 2004).

4.3.2 A Context Graph Kernel

4.3.2.1 Kernel Design

Recognizing the limitations of previous research, I adopt a context assumption and use multiple levels of gene interactions for gene function prediction. As shown in Figure 4.1d, I represent each gene's context as a graph. A context graph centers on the focal gene and contains its direct and indirect neighbors. According to the context assumption, genes with similar context graphs may share similar functions. Therefore I design a context graph kernel (CGK) to compute the similarity between context graphs.

Similar to the labeled graph kernel proposed in the last chapter, the proposed CGK also relies on the comparisons of random walk paths in the graphs. It belongs to the family of convolution kernels (Haussler, 1999) and computes the similarity between graphs by accumulating the similarity scores of random walk paths on the graphs in a pairwise manner. The CGK considers the random walk paths that start from the focal gene. These paths represent the gene pathways related to the focal gene and thus may potentially indicate its functions. I calculate the similarity between two context graphs as the sum of pairwise similarities of these random walk paths. Each path's contribution is weighted according to its probability to appear among all paths, i.e., probability of existence. Since longer random walk paths have a relatively lower probability of existence, the genes that are far from the focal gene have less impact on the focal gene's function.

The following procedure summarizes the kernel design:

(i) In the gene interaction network G of a genome with n genes $\{g_1, g_2, \ldots g_n\}$, I represent gene g_x's context graph as G_x. All random walks in G_x start at g_x. At each gene (node) g_i, a random walk has a probability of $p_s(g_i)$ to stop and a probability of $p_t(g_j|g_i)$ to jump to one of g_i's neighbors, g_j. Thus, a random walk path of length l, $h=(g_{<h,0>}\rightarrow g_{<h,1>}\rightarrow\ldots\rightarrow g_{<h,l>})$, has the probability of existence:

$$P(h\,|\,G_x) = p_t(g_{<h,1>}\,|\,g_{<h,0>})\,p_t(g_{<h,2>}\,|\,g_{<h,1>})\cdots p_t(g_{<h,l>}\,|\,g_{<h,l-1>})\,p_s(g_l)$$

where $g_{<h,i>}$ indicates the i-th gene on the path h.

(ii) After enumerating all random walk paths, the similarity between two context graphs, $K(G_x,G_y)$, is defined as the sum of the similarity scores of random walk paths weighted by the paths' probability of existence:

$$K(G_x,G_y) = \sum_{h_i\in H(G_x)}\ \sum_{h_j\in H(Gy)} K_h(h_i,h_j)P(h_i\,|\,G_x)P(h_j\,|\,G_y)$$

where $K_h(h_i,h_j)$ is the similarity score between two random walk paths h_i and h_j, and $H(G_x)$ and $H(G_y)$ denote the sets of random walk paths in the two context graphs.

(iii) The similarity between random walk paths is computed by multiplying the similarity of the corresponding nodes along the two paths. I define $K_h(h_i,h_j)$ as:

$$K_h(h_i,h_j) = \begin{cases} \prod_k K_g(g_{<h_i,k>}, g_{<h_j,k>}, & if\ |h_i|=|h_j| \\ 0, & otherwise \end{cases}$$

where $|h_i|$ and $|h_j|$ are the lengths of the two paths h_i and h_j, and $K_g(g_{<h_i,k>}, g_{<h_j,k>})$ is a similarity function defined on nodes.

(iv) The node similarity $K_g(g_{<h_i,k>}, g_{<h_j,k>})$ uses only the information of individual genes, potentially available from various biological data sources such as experiments, literature, gene sequences, and ontologies. For instance, known gene functions are often used as evidence for interaction-based function prediction (Hishigaki et al., 2001; Vazquez et al., 2003; Nabieva et al., 2005; Murali et al., 2006). Gene functions are often defined as a hierarchical structure in ontologies such as Gene Ontology (GO) (Ashburner et al., 2000), thus the similarity between two function labels can be defined based on the number of their common ancestors in the hierarchy. Since a gene may have multiple known function labels, the similarity between two genes can be calculated by summing up the pairwise similarity scores of all their functions:

$$K_f(g_i, g_j) = \sum_{l_i} \sum_{l_j} (1 - \rho^{common(l_i, l_j)})$$

where l_i and l_j are the known functions of g_i and g_j, and common(l_i, l_j) is the number of the two functions' common ancestors. $1-\rho^{common(l_i,l_j)}$ is the sum of functional similarities for each level of common ancestors (i.e., $\sum_{k=1}^{common(l_i,l_j)} \left[(1-\rho)\cdot\rho^{k-1}\right]$), where higher level ancestors have higher weight than lower level ones by a damping fact ρ (0<ρ<1) between each level.

(v) In practice, kernel normalization may be considered for better classification performance on some datasets:

$$K'(G_x, G_y) = K(G_x, G_y) / \sqrt{K(G_x, G_x) K(G_y, G_y)}\,.$$

4.3.2.2 Computing the Context Graph Kernel in a Matrix Form

To compute the CGK by enumerating all random walk paths is computationally expensive. In this research, I introduce an efficient method for computing the CGK based on the matrix form of the kernel.

In the gene interaction network G of a genome with n genes $\{g_1, g_2, \dots g_n\}$, I use two matrices $M=\{M_{i,j}\}=\{p_t(g_j|g_i)\}$ and $Q=\{Q_{i,j}\}=\{p_s(g_i)\}$ to encode each node's transition probability and stopping probability in the graph, respectively. The context graph kernel matrix of the entire genome $\tilde{K} = \{\tilde{K}_{i,j}\} = \{K(G_i, G_j)\}$ can be represented as the summation of a series of matrices (Proposition in Appendix A):

$$\tilde{K} = \sum_i K_i \qquad (i=1,2,\dots,\infty)$$

where $K_1=(M*Q)K_0(M*Q)^T$, $K_{i+1}=M(K_0*K_i)M^T$ (i=1,2,…,∞) , and $K_0 = \{K_{0_{i,j}}\} = \{K_g(g_i, g_j)\}$ is the kernel matrix of node information. The operation * is the Hadamard product (i.e., entrywise product) where A*B=$\{a_{i,j} \cdot b_{i,j}\}$.

Each matrix K_i covers the random walk paths of length = i in the context graphs. It can be proved that when r approaches +∞, $K_1+K_2+\dots+K_r$ converges if there is a normalized K_0 and the stop probability $p_s(g_i)$ is unified or larger than 0.5 (Proposition in Appendix B). Therefore, given a maximum length of random walk paths, r, we can use $K_1+K_2+\dots+K_r$ to approximate the kernel $\tilde{K}$. With the decomposed form of K_i, the kernel $\tilde{K}$ can be computed by simple matrix operations with a time complexity in $O(rn^3)$, where n is the number of nodes in the network.

Another advantage of the matrix formulation is that it facilitates the investigation of the effect of indirect interactions on gene function prediction. Since all random walks start from the focal genes, each K_i covers the genes that are i step(s) away from the focal gene. While $\sum_{i=1}^{r} K_i$ will converge as r increases, specifying a different r restricts the CGK to a limited number of indirect interactions and may yield a different prediction performance. It may help us understand the effect of indirect interactions in gene function prediction.

4.4 Experimental Study

4.4.1 Dataset

4.4.1.1 Human Genome Gene Interaction Network

In this study I used the collection of gene interactions from the BioGRID database (Breitkreutz et al., 2003) to construct a gene interaction network of human genomes. BioGRID is a free and well-known database with protein/gene interactions manually curated from Medline literature. I extracted 38,225 relations related to Homo sapiens genes from BioGRID (version 2.026). By mapping proteins to genes and consolidating duplicate relations, I constructed a gene interaction network with 19,623 non-directional relations among 7,167 genes.

4.4.1.2 Gene Function Labels

Following previous studies and domain experts' suggestions, I used terms from the "biological process" hierarchy of Gene Ontology (Ashburner et al., 2000) as gene function labels. The "biological process" hierarchy is a 10-level structure of 7,172 GO terms in the dataset (downloaded in 2003). It has 7 second-level terms (including one "unknown" class) and 264 third-level terms. I use the third-level terms as class labels in the study so as to have both enough classification granularity and sufficient training/testing data instances for each class. The genes whose functions are not documented in GO were annotated as "unknown."

4.4.1.3 A P53-related Testbed

The tumor suppressor gene, p53, plays a central role in the regulation of apoptosis and cell cycle arrest in cancer development. P53-related genes have attracted much attention and their functions are well-studied as compared to other human genes. Therefore, I chose p53-related genes as the research testbed. In chapter 2, I identified 2,045 p53-related genes from the Medline abstracts with a Natural Language Processing tool. After eliminating the genes without known functions, I had 1,436 genes within 38 "biological process" functions. Nine functions that have more than 50 instances are used in the experiments for evaluation (Table 4.2). While each of the genes is a training/testing instance in the experiment, I used the entire human genome gene interaction network to extract context graphs and generate features for function prediction.

Table 4.2 Nine Major Gene Functions in the P53 Testbed

GO term	Description	Parent GO term	Number of instances
GO:6928	cell motility	GO:9987 (cellular process)	66
GO:30154	cell differentiation	GO:9987 (cellular process); GO:7275 (development)	71
GO:16265	death	GO:7582 (physiological processes)	181
GO:9653	morphogenesis	GO:7275 (development)	200
GO:9605	response to external stimulus	GO:7582 (physiological processes)	214
GO:6950	response to stress	GO:7582 (physiological processes)	220
GO:8151	cell growth and/ or maintenance	GO:9987 (cellular process); GO:7582 (physiological processes)	499
GO:7154	cell communication	GO:9987 (cellular process)	663
GO:8152	metabolism	GO:7582 (physiological processes)	854

4.4.2 Experimental Procedures

While implementing the CGK, I used the known gene functions as node information because they have been often used as evidence for function prediction. The "unknown" functions of the genes are considered as having the same probability of being in any second-level GO categories for kernel calculation. When generating random walks on the gene interaction network G, I specified a uniform stopping probability $p_s(g_i)$=1-λ (0<λ<1) for all random walks for the sake of simplicity. I also assumed equal probability of jumping from one node to any of its neighbors, i.e., $p_t(g_j|g_i)=\lambda/d(g_i)$, where $d(g_i)$ is the number of g_i's interacting genes. I used a popular SVM package, libSVM (Chang and Lin, 2001), to build classifiers based on the CGK kernel. The CGK kernel was normalized before being fed into libSVM.

To test the algorithm's performance, I conducted 10-fold cross-validations on the p53 dataset. In each run, the functions of genes in the testing fold were treated as "unknown" in the gene interaction network. To specify λ, ρ (for node information), and the parameters of the SVM algorithm, I used one fold of the data and utilized the tool provided by libSVM to select parameters based on the number of correctly predicted genes. The parameter setting optimized for this one fold of data were later applied on the entire dataset for evaluation. In the experiments, ρ was finalized as 0.9 (from 0.5 to 0.9) and λ was selected as 0.8 (from 0.1 to 0.9) from this tuning process.

For the study, I designed two sets of experiments: 1) to examine the effect of indirect interactions in gene function prediction and 2) to compare the proposed CGK-based method with other state-of-the-art methods.

As described in Section 4.3.2.2, using $K_1+K_2+\ldots+K_r$ to approximate the kernel $\tilde{K}$ restricts the CGK to using the indirect interactions that are up to r steps away from the focal genes. In addition, the stopping probability, $p_s(g_i)$, also affects the use of indirect interactions by controlling the probability of longer random walks' appearance. Thus, in the first experiment, I compared the performances of CGK based on different *r* and λ settings to inspect the effect of indirect interactions.

For the second experiment, I compared the CGK with four baseline methods from previous studies. In this experiment, I specified λ=0.8 based on the tuning process and calculated the CGK kernel matrix till convergence. The same 10-fold validation method was applied to all prediction models in this experiment for a fair comparison.

4.4.3 Evaluation Metrics

I evaluated the performance of the classification models using precision, recall, and F-measure, which are common evaluation metrics in gene function prediction studies (Karaoz et al., 2004; Murali et al., 2006; Sharan et al., 2007). Since one gene may have more than one function and one function may be associated with more than one gene, I calculated the three measures at both the instance level (i.e., gene level) and class level (i.e., function level). I also inspected instance-level performance with respect to the number of interacting genes to better understand the algorithms' characteristics (Hishigaki et al., 2001).

Instance-level precision P_i, recall R_i, and F-measure F_i are defined as:

$$P_i = \frac{\text{correctly predicted functions of a gene}}{\text{all predicted functions of a gene}}$$

$$R_i = \frac{\text{correctly predicted functions of a gene}}{\text{all (known) functions of a gene}}$$

$$F_i = 2 \times P_i \times R_i / (P_i + R_i)$$

Class-level precision P_c, recall R_c, and F-measure F_c are defined as:

$$P_c = \frac{\text{correctly predicted genes of a class}}{\text{all predicted genes of a class}}$$

$$R_c = \frac{\text{correctly predicted genes of a class}}{\text{all (known) genes of a class}}$$

$$F_c = 2 \times P_c \times R_c / (P_c + R_c)$$

4.5 Results and Discussion

4.5.1 Experiment I: Effect of Indirect Interactions

Figure 4.3 shows the instance-level and class-level performances using different *r* and λ. When only direct interactions are considered (*r*=1), the normalized CGK kernels are the same for different λ, which achieves average F-measure scores of 59.2% at the instance level and 33.3% at the class level. When additional levels of indirect interactions are taken into account, for most λ settings the average F-measure score curves show an increasing trend until convergence (for λ=0.8 and λ=0.9, a little fluctuation is shown for instance-level F-measure). In addition, when I specify parameter *r*, a larger λ tends to lead to a better performance. For the instance-level performance, the performance curves converge to similar values for λ=0.6 and λ=0.8.

It should be noted that a larger *r* indicates more indirect interaction can be used in the experiments. A larger λ indicates that indirect interactions have a larger probability to be adopted by the kernel. The experimental results on *r* and λ demonstrate that incorporating information of indirect interactions in context-based models can improve the gene function prediction performance. The convergence of the performance curves when r is larger is expected as the kernel's convergence property (Proposition in Appendix B). Moreover, the experiments suggest that computing CGK up to two to three levels (r = 2 or 3) gives a sufficiently accurate approximation of the CGK, which can be used in future empirical studies.

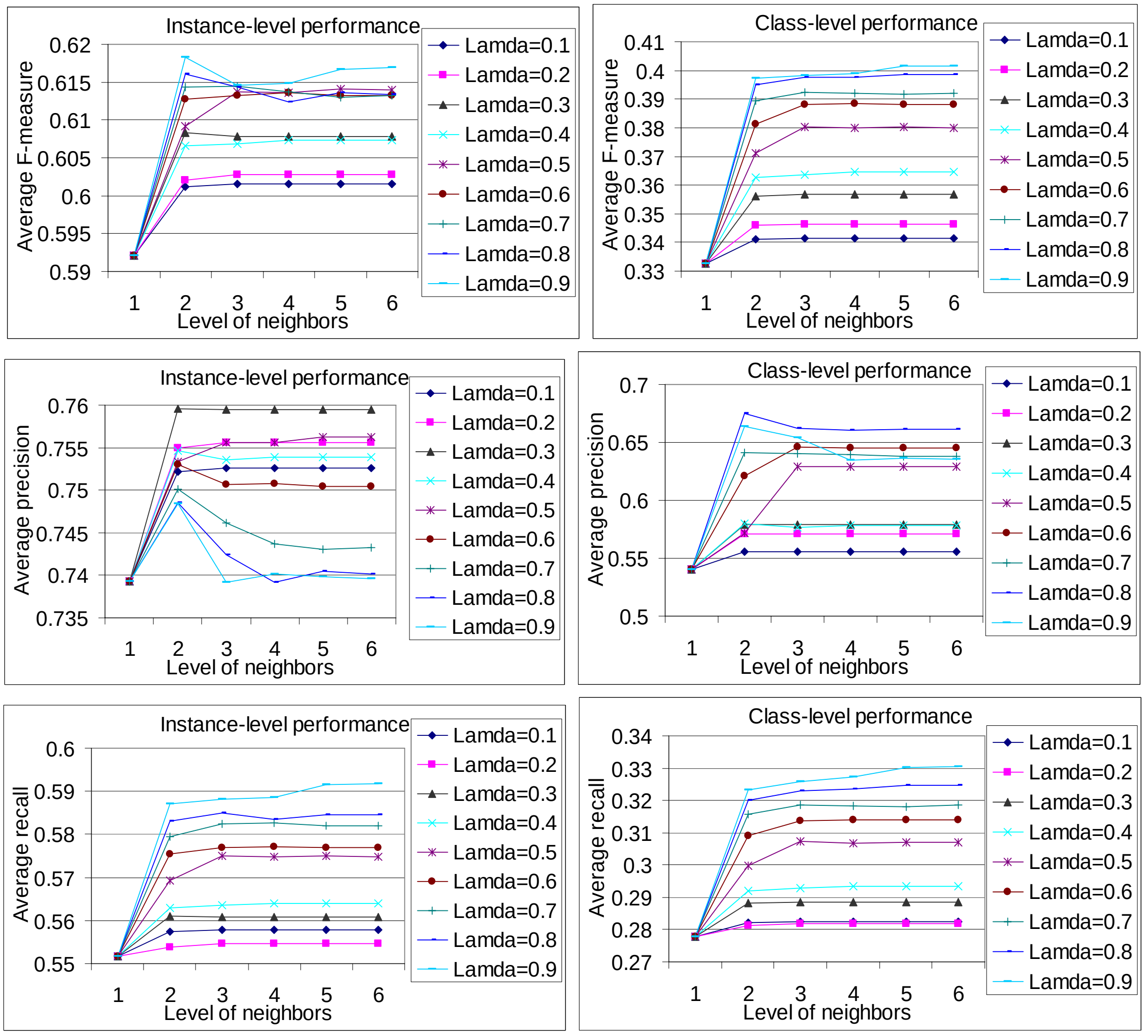


Figure 4.3 Performance of CGK Using Different Levels of Interactions

4.5.2 Experiment II: CGK vs. Other Methods

4.5.2.1 Benchmark Algorithms

Based on the three dimensions considered in the literature review (i.e., assumption, interactions, and techniques), I compare the proposed context graph kernel method against four baseline methods from previous studies (Table 4.3):

Table 4.3 Baseline Methods for Gene Function Prediction

Benchmark algorithms	Assumptions	Interactions	Techniques
Linear kernel	Context	Direct	Learning
Diffusion kernels	Linkage	Multi-level	Learning
Gene Annotation using Interaction Network	Linkage	Multi-level	Heuristic
Majority voting	Linkage	Direct	Heuristic

a) Linear kernel: A linear kernel (LK) uses direct interactions under a context assumption (Lanckriet et al., 2004). For gene function prediction, the genes directly connected to the focal gene are represented as the gene's features. The inner product of the feature vectors is used to calculate genes' similarities. Genes with higher similarities, i.e., genes sharing a larger number of neighbor genes, are predicted to have similar functions.

b) Diffusion kernel: A diffusion kernel (DK) uses multi-level interactions under a linkage assumption. DK uses genes' relative positions in the gene interaction network to predict their functions. Two genes that have more and shorter paths between them are more likely to be predicted to share function labels (Lanckriet et al., 2004; Yamanishi et al., 2004). In addition to using the traditional diffusion kernel, I also adopt a locally constrained diffusion kernel (LDK) as proposed by Tsuda et al. due to its reported superior performance (Tsuda and Noble, 2004).

c) Gene Annotation using Interaction Network (GAIN): GAIN is a heuristic method using gene interaction networks under a linkage assumption, as reported in Karaoz et al. (Karaoz et al., 2004) and Murali et al. (Murali et al., 2006). It optimizes gene function assignment by minimizing the inconsistency among connected genes in the network.

d) Neighbor majority voting method (MV): Guided by the "guilt by association" rule (Mayer and Hieter, 2000), I construct a simple neighbor majority voting classifier. The most dominant function label in the directly connected neighbors is predicted as the focal gene's function.

In this research the implementation of LDK and GAIN were provided by the authors, while the others were implemented by us. For the context graph kernel, I adopt the parameter setting from the tuning process explained before. I calculate the kernel till convergence (*r*=6).

4.5.2.2 Instance-level Performance

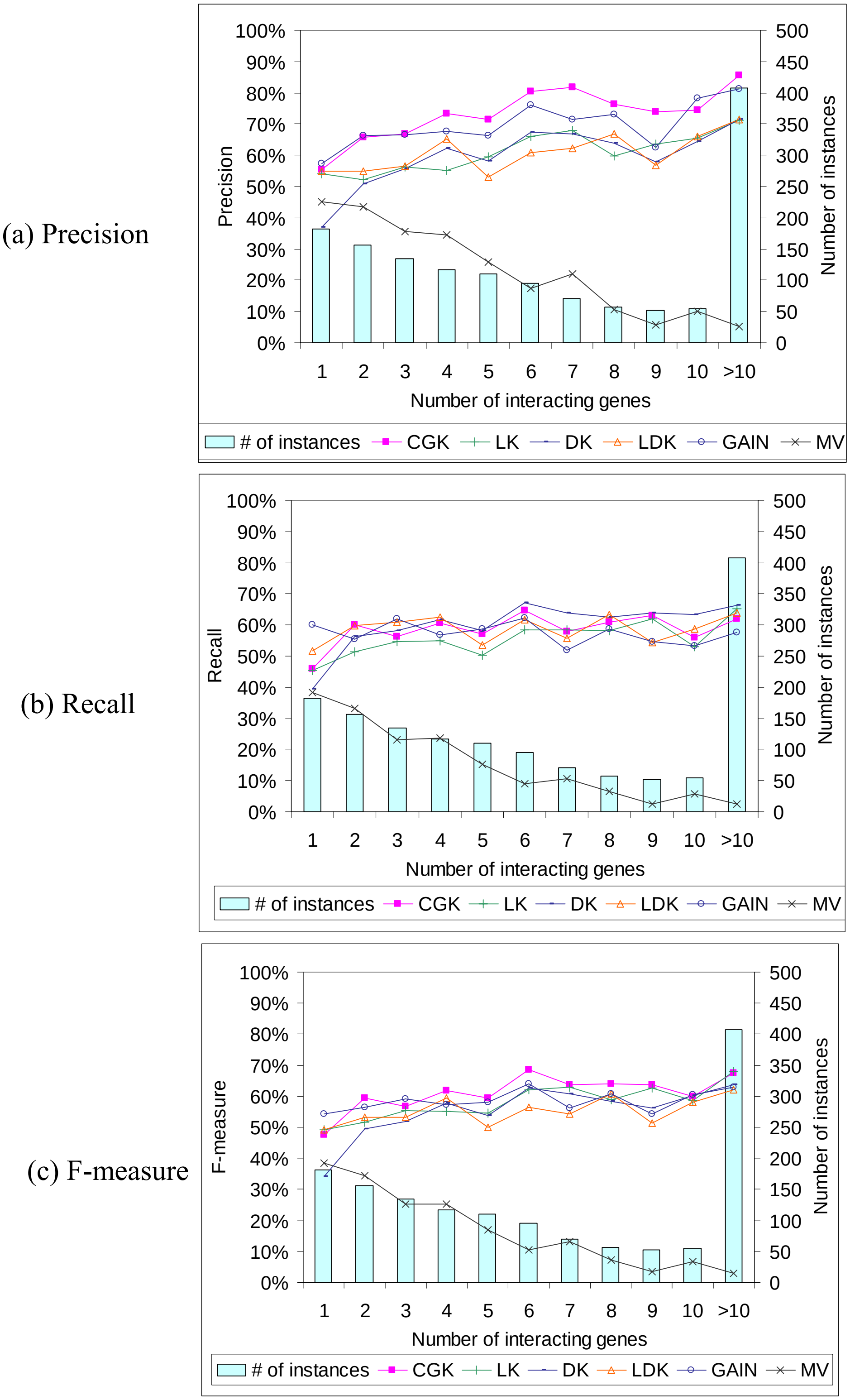
Table 4.4 shows the instance-level prediction performances for different methods. The proposed context graph kernel achieves the highest average precision (74.01%) and F-measure (61.34%), which are significantly better than other methods with a p value < 0.0001 in pairwise t-tests. The DK achieves the highest recall but not significantly different from that of CGK's (p value>0.12). Among all learning-based methods, the context graph kernel outperforms the linear kernel by about 12% and the diffusion kernels by about 12%~14% in precision. Overall, the context graph kernel provides more precise predictions without missing significantly more hits than other methods. In these experiments, the majority voting algorithm has significantly worse performance than the other algorithms, due to its use of only direct interactions. This finding is consistent with previous research (Vazquez et al., 2003).

Table 4.4 Instance-level Prediction Performance

Instance-level performance	Average precision	Average recall	Average F-measure
Context Graph Kernel (CGK)	**74.01%**	**58.45%**	**61.34%**
Linear Kernel (LK)	61.94%	56.55%	55.00%
Diffusion Kernel (DK)	60.08%	**59.63%**	55.24%
Locally Constrained Diffusion Kernel (LDK)	62.18%	**59.58%**	56.21%
Gene Annotation using Interaction Network (GAIN)	71.08%	57.91%	59.18%
Majority Voting (MV)	23.33%	16.07%	17.20%

* Within the same measure, the bold numbers do not have a significant difference from the largest one at the 90% confidence level.

Figure 4.4 shows different classifiers' performances for genes with different numbers of interacting genes (or in graph theoretical terms, nodes with different degrees). The number of genes in each group is shown by column bars. In general, there is a positive correlation between classification performance and the number of interacting genes, except for the majority voting algorithm. Most algorithms were able to capture the information provided by a larger number of interactions for a more accurate prediction. Figure 4 also shows that the CGK performance is not the best for genes with a smaller number of interacting genes. However, as the number of interacting genes increases, CGK's performance becomes more competitive.



(a) Precision

(b) Recall

(c) F-measure

Figure 4.4 Instance-level Performance for Genes with Different Interacting Genes

4.5.2.3 Class-level Performance

Table 4.5 shows the class-level performances of different classifiers. For most classes, the context graph kernel achieves the highest precision, while the two diffusion kernels (DK and LDK) achieve the highest recall. Both CGK and the diffusion kernels use a machine learning approach and utilize the structure of the gene interaction network. Their performance differences show the different prediction power of the context assumption and the linkage assumption.

Table 4.5 Class-level Prediction Performances

GO term		6928	30154	16265	9653	9605	6950	8151	7154	8152
# of genes		66	71	181	200	214	220	499	663	854
CGK	P_c	25.00%	75.00%	81.58%	50.68%	62.92%	76.12%	62.59%	78.55%	82.67%
	R_c	1.52%	4.23%	34.25%	18.50%	26.17%	23.18%	35.87%	70.14%	78.22%
	F_c	2.86%	8.00%	48.25%	27.11%	36.96%	35.54%	45.61%	74.10%	80.39%
LK	P_c	20.00%	29.03%	48.28%	26.72%	40.00%	40.74%	48.64%	73.36%	73.43%
	R_c	4.55%	12.68%	30.94%	17.50%	28.97%	25.00%	39.48%	65.61%	79.63%
	F_c	7.41%	17.65%	37.71%	21.15%	33.60%	30.99%	43.58%	69.27%	76.40%
DK	P_c	16.22%	27.08%	52.08%	35.00%	38.17%	38.20%	48.54%	70.45%	73.15%
	R_c	9.09%	18.31%	41.44%	28.00%	33.18%	30.91%	46.69%	70.14%	74.00%
	F_c	11.65%	21.85%	46.15%	31.11%	35.50%	34.17%	47.60%	70.29%	73.57%
LDK	P_c	10.00%	28.21%	56.49%	32.56%	43.82%	41.28%	50.00%	71.97%	72.31%
	R_c	6.06%	15.49%	40.88%	21.00%	36.45%	32.27%	41.88%	68.17%	77.99%
	F_c	7.55%	20.00%	47.44%	25.53%	39.80%	36.22%	45.58%	70.02%	75.04%
GAIN	P_c	12.50%	38.46%	70.89%	46.94%	68.75%	48.42%	57.53%	75.24%	74.83%
	R_c	3.03%	7.04%	30.94%	11.50%	25.70%	20.91%	38.28%	72.40%	76.23%
	F_c	4.88%	11.90%	43.08%	18.47%	37.41%	29.21%	45.97%	73.79%	75.52%
MV	P_c	25.87%	29.19%	11.50%	13.82%	12.64%	11.94%	40.54%	47.17%	65.00%
	R_c	16.82%	21.96%	7.18%	8.50%	16.67%	11.27%	15.03%	11.31%	15.22%
	F_c	20.39%	25.07%	8.84%	10.53%	14.38%	11.59%	21.93%	18.25%	24.67%

From the experimental results, I observe a positive correlation between the class-level performance measures and the number of instances in the classes. I also identify some classes that are difficult to classify (compared with classes with a similar number of instances), such as “GO:0009653 (morphogenesis)” and “GO:0006928 (cell motility).” Both of these classes belong to a super class “GO:0007275 (development)” in the GO ontology. Function prediction methods based on gene interactions do not perform well on these two classes, probably because their functions are less related to gene interactions. Other types of information may need to be considered in this case.

4.6 Summary

In this research, I propose a context graph kernel for gene function prediction. This approach is based on a context assumption and leverages multiple levels of interactions in gene interaction networks. The experiments on a testbed of p53-related genes show that the prediction performance of the CGK increases when more levels of indirect interactions are considered. When compared to other state-of-the-art methods that often use linkage assumptions and/or direct interactions, the proposed approach is highly competitive and achieves the highest F-measure (61.34%). In addition, I find that the proposed approach works better on genes with a larger number of interacting genes and function classes with a larger number of genes.

The context graph kernel extends the labeled graph kernel and enables us to incorporate different types of node information in the node classification task. Together they show the effectiveness of using graph structure in the task of analyzing individual nodes. In the next chapter, I will explore using the graph structure to analyze relations between nodes, i.e., a link prediction task in knowledge discovery.

CHAPTER 5. LINK PREDICTION: ADDRESSING THE RECOMMENDATION PROBLEM WITHIN BIPARTITE GRAPHS

5.1 Introduction

In the previous two chapters, I have studied the node classification problem. In this chapter, I study the relationship between entity pairs in graph-structured data, i.e., the link prediction task. Specifically, I examine this with an example application in recommender systems.

Recommender systems are widely used in recommending products, services, and contents to users. As the core of recommender systems, recommendation algorithms usually rely on user characteristics, item attributes, and user-item interactions (browsing, rating, purchasing, etc.) to infer user interests (Pazzani, 1999). Successful collaborative filtering (CF) algorithms take advantage of user-item interactions and infer users' interests based on their overlapping usage history.

In CF, users (or customers) and items (or products) can be considered as nodes in a bipartite graph linked by their interactions. Such a representation converts the recommendation problem into a link prediction problem. The success of previous CF algorithms suggests that graph-related features from the user-item graph may be useful in the recommendation process. However, such graph-related features were mainly used in heuristic algorithms (Huang et al., 2004b; Zhou et al., 2007). The use of graph-related features in learning-based studies is still limited. In addition, the few existing learning-based recommendation algorithms usually rely on explicit feature extraction, which needs intensive domain knowledge and is computationally expensive. These disadvantages limited the use of graph information in learning-based recommendation algorithms.

In this chapter, I propose a kernel-based machine learning approach to address the recommendation problem as a link prediction problem in user-item interaction graphs. I study an associative interaction graph (AIG) for each user-item pair, and develop a graph kernel that captures the features in the AIGs to predict if a link may exist between the user-item pair or not. I demonstrate the improved recommendation performance of the new method using three real-world datasets.

This chapter is organized as follows. Section 5.2 reviews related studies on recommendation algorithms. Section 5.3 introduces the proposed graph kernel-based recommendation framework. Section 5.4 describes the experiments on three real-world datasets. Section 5.5 discusses the experimental results. Section 5.6 summarizes the findings.

5.2 Literature Review

5.2.1 Recommendation Algorithms

Huang et al. (Huang et al., 2004c) and Adomavicius et al. (Adomavicius and Tuzhilin, 2005) have conducted comprehensive reviews on recommender systems and recommendation algorithm studies. In general, recommender system research can be characterized by their recommendation algorithms, system inputs (i.e., user information (Pazzani, 1999), item information (Linden et al., 2003), transaction information (Huang et al., 2004d)), and system outputs (i.e., whether the predictions are linkages/purchases, or numerical ratings).

Focusing on the recommendation algorithm design, I review previous studies according to their feature types and computational techniques.

5.2.1.1 Feature Types

According to the feature type, previous recommendation algorithm studies fall into two groups: the ones using local features and the ones using graph-related features.

1) Local features

Local features include both individual user/items' features and the statistical characteristics of user usage histories. The local features have been used to define user/item similarity measures, based on which similar products can be cross-recommended between similar users (Ahn, 2008). The statistical patterns of users' purchasing histories can be captured by probabilistic models. Such models may introduce hidden classes of users/items and estimate user-item pairs' probabilities of having interactions given existing transactions (Hofmann, 2004).

2) Graph-related features

Graph-related features are defined on graphs constructed from user-item interactions based on graph theory and topology analysis. User-item interactions can be represented as bipartite graphs, where users and items are nodes and business transactions are links. Eigenvector-based node ranking algorithms that are similar to PageRank (Huang et al., 2004b; Griffith et al., 2006; Gori and Pucci, 2007) and HITS (Huang et al., 2007b) have been applied on such graphs to rank items and recommend highly ranked items to the users. In addition, some research has projected the bipartite user-item graph onto a unipartite user (or item) graph (Zhou et al., 2007) and derived user (item) similarities using graph-based algorithms. The similarity measures can be used to cross-recommend items between users. Moreover, the graphs can be constructed from user/item characteristics other than transactions. For example, graphs reflecting similarities of user usage logs and item contents were used in previous research (Bollen et al., 2007).

5.2.1.2 Computational Techniques

According to computational techniques, previous recommendation algorithms can be classified as heuristics and learning-based algorithms.

1) Heuristics

Heuristic methods use predefined measures or rules to rank items and make recommendations. For example, recommending the most popular items to every user is the simplest heuristic that can be applied. Similarity-based heuristics that recommend similar items between similar users are another type of heuristic algorithm. The similarity measures can be defined on users (using demographic information or purchase history information) or items (using item content, attribute, specification, or sales information). In previous research, several similarity measures have been proposed (Ahn, 2008), while the most popular measures were Pearson correlation coefficient (Resnick et al., 1994) and cosine-based similarity (Sarwar et al., 2001).

Heuristic methods can also be designed based on graph-related features. In addition to the popular eigenvector-based node ranking algorithms (Huang et al., 2007b), some studies have focused on the positional information of users and items in their interaction networks. Fouss et al. proposed that closer users and items (as measured by conducting random walks on the network) may have a higher probability of interacting

(Fouss et al., 2007). Moreover, Huang et al. proposed to recommend the links that can lead to higher graph clustering coefficients (Huang et al., 2007a).

2) Learning-based methods

Learning-based methods build models on the data for future recommendations. Focusing on the transactions in user purchase history, some studies have built probabilistic models to uncover the hidden class (Polcicova and Tino, 2004; Yu et al., 2004; Zeng et al., 2004) among users and items, as well as to interpret user-item relationships. For instance, Hofmann et al. applied the probabilistic latent semantic analysis (PLSA) model in recommendation (Hofmann, 2004). Iwata et al. applied a maximum entropy model to better capture the temporal information in user purchase histories for recommendation (Iwata et al., 2008). Another approach of learning-based methods was to take advantage of mature models and explore feature construction methods that can achieve better recommendation performance. The probabilistic relational model (PRM) (Getoor and Sahami, 1999; Newton and Greiner, 2004), the regression model (Vucetic and Obradovic, 2005), and the SVM algorithm (Xu and Araki, 2006) have all been applied to the recommendation problem.

Considering business transactions as a user-item graph, the graph-related features have been shown to be beneficial in learning-based recommendation algorithms. Huang et al. extended the PRM framework (Huang et al., 2004d) and proposed to take advantage of features from directly or indirectly connected nodes for recommendation. Yajima used the Laplacian kernel to capture the distances between items and build one-class SVM recommendation models to predict positionally closer items for each user (Yajima, 2006).

Reddy et al. used a graph-based clustering algorithm to find similar users in user-item graphs for the purpose of recommendation (Reddy et al., 2002). Hasan et al. proposed a general supervised framework to use proximity features, aggregated features, and topological features for link prediction, which can be utilized in the recommendation problem (Hasan et al., 2006).

5.2.2 Research Gaps

Table 5.1 summarizes previous research on recommendation algorithms. While several efforts have been made to use graph-related features in heuristics, the heuristic algorithms' performance varies with datasets, due to the fact that the models are not built based on the data. Compared to heuristic algorithms, the studies on learning-based recommendation algorithms that used graph-related features are still limited. The few existing learning-based algorithms in that category, such as (Huang et al., 2004d), usually need explicitly defined features, which require intensive domain knowledge. In addition, it is computationally expensive to specify the graph-related features in user-item interaction networks, due to the fan-out characteristic of graph-structured data.

Table 5.1 A Summary of Previous Recommendation Algorithm Studies

Studies	Feature*	Technique#	Notes
(Resnick et al., 1994)	L	H	Similarity-based
(Sarwar et al., 2001)	L	H	
(Ahn, 2008)	L	H	
(Huang et al., 2004b)	G	H	Eigenvector-based node ranking
(Griffith et al., 2006)	G	H	
(Gori and Pucci, 2007)	G	H	
(Huang et al., 2007b)	G	H	
(Zhou et al., 2007)	G	H	
(Fouss et al., 2007)	G	H	Node position-based
(Huang et al., 2007a)	G	H	Clustering coefficient-based
(Hofmann, 2004)	L	L	Probabilistic model
(Yu et al., 2004)	L	L	
(Zeng et al., 2004)	L	L	
(Polcicova and Tino, 2004)	L	L	
(Iwata et al., 2008)	L	L	
(Getoor and Sahami, 1999)	L	L	Mature model + local features
(Newton and Greiner, 2004)	L	L	
(Vucetic and Obradovic, 2005)	L	L	
(Adomavicius and Tuzhilin, 2005)	L	L	
(Xu and Araki, 2006)	L	L	
(Huang et al., 2004d)	G	L	PRM + aggregative features
(Yajima, 2006)	G	L	One-class SVM + Laplacian kernel
(Reddy et al., 2002)	G	L	Graph-based clustering

* L – local features; G – graph-related features.
# H – heuristic methods; L – learning-based methods.

5.3 Research Design

In this research, I take a learning-based approach to explore graph-related features for making recommendations. I adopt a kernel-based framework due to its ability to incorporate structural information without enumerating features in the learning process.

To predict whether a user-item pair may interact, I examine the user-item interactions that are related to the focal user-item pair. From a graph perspective, I define an AIG for each user-item pair. The AIG includes the users and items that are *n* steps from the focal pair and their interactions. The topologically similar AIGs may include users with similar usage behaviors and items with similar access characteristics. Thus, if two user-item pairs have similar AIGs, they may have a similar probability of having interactions in the future. In a kernel-based framework, this design is implemented by designing a graph kernel to model the AIGs to classify user-item pairs as possible or impossible links.

5.3.1 A Graph Kernel-based Recommendation Framework

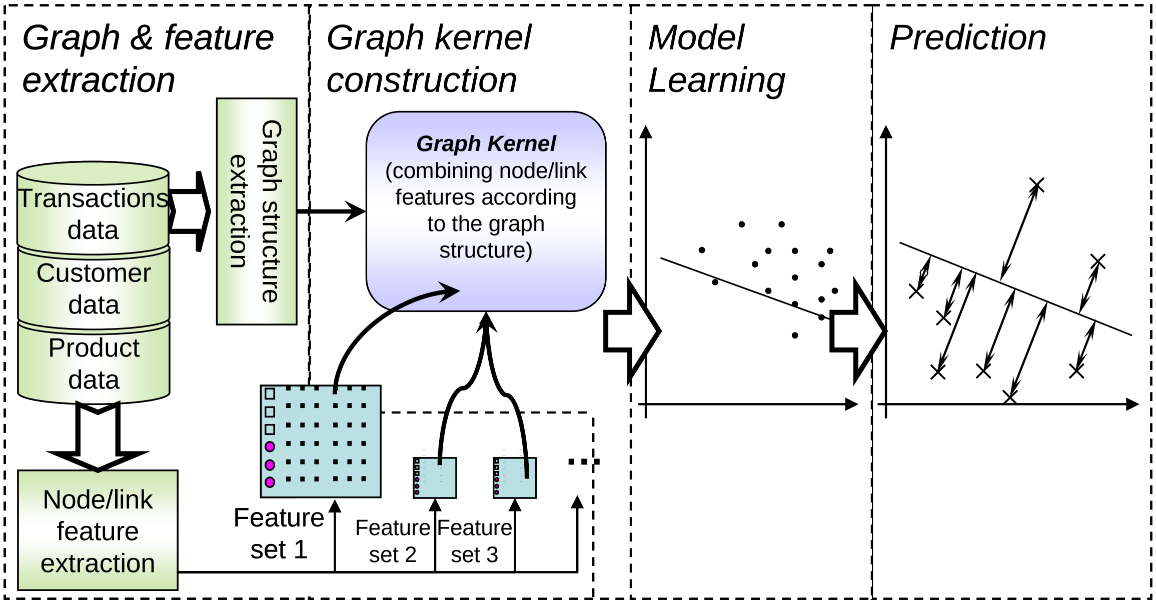

Graph & feature extraction
Graph kernel construction
Model Learning
Prediction
Transactions data
Customer data
Product data
Graph structure extraction
Graph Kernel (combining node/link features according to the graph structure)
Node/link feature extraction
Feature set 1
Feature set 2
Feature set 3

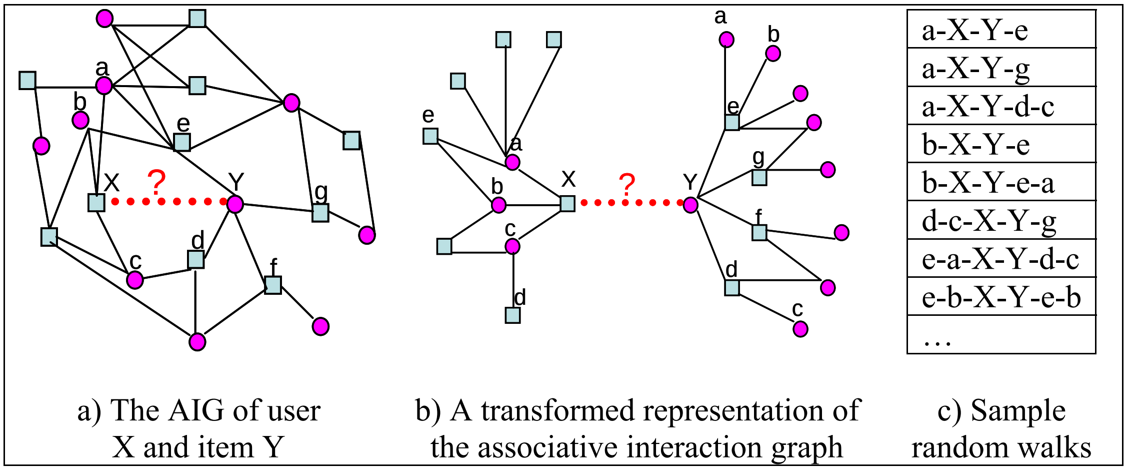
Figure 5.1 A Graph Kernel-based Recommendation Framework

Figure 5.1 shows the four steps in the graph kernel-based framework. 1) In the graph and feature extraction step, I construct the user-item graph from user usage histories. I also extract features describing nodes (users and items) and links (transaction information) from the data. 2) In the graph kernel construction step, I define a graph similarity measure that can capture the structure of AIGs. This kernel design is the most essential module in kernel methods, which I will elaborate on later. 3) In the model learning step, a binary classifier is built to separate potential links from impossible links. In the recommendation problem, all known data instances are the interactions that have happened. There are no negative data instances. Thus, I adopt a one-class SVM algorithm (Scholkopf et al., 1999), which looks for a hyper-plane to separate positive data instances from the origin. In this algorithm, the unclassified data instances on the origin side of the hyper-plane are predicted to be negative (i.e., impossible to exist). 4) In the prediction step, it is necessary to provide prediction confidence values to rank the items for recommendation. I use the Euclid distance between a data instance (i.e., user-item pair) and the classification hyper-plane as the confidence estimation. Such a measure is proportional to several classification confidence probability estimation measures for SVM (Wu et al., 2004). Since recommendation algorithms only need ranks of items, this kind of method is sufficient for this research.

5.3.2 Graph Kernel Design

The graph kernel defines a similarity measure for the AIGs. I decompose graphs to random walk paths to measure graph similarities. The decomposition method has been shown to be effective in previous research (Borgwardt et al., 2005; Li et al., 2007b). Moreover, I consider only the random walk paths that pass through the hypothetical link of the focal user-item pair in each AIG. In such a design the nodes (users and items) closer to the focal pair are more likely to be used for prediction. The graph kernel is calculated as the summation of pairwise comparisons of matching random walk paths weighted by their probability of existence.

Figure 5.2a shows an example of a user-item pair (X-Y) and its AIG. In Figure 5.2b, I present a transformed representation of the AIG to show how random walks passing through the hypothetical link of X-Y can be generated. It should be noted that some nodes are presented more than once for explanation. For example, the node “a” that is one step away from X on the left side is identical to the node “a” that is two steps away from Y on the right side. Examples of the random walk paths are shown in Figure 5.2c. These random walk paths can also be considered to be starting from X and Y and going to the other parts of the graph simultaneously.



a-X-Y-e
a-X-Y-g
a-X-Y-d-c
b-X-Y-e
b-X-Y-e-a
d-c-X-Y-g
e-a-X-Y-d-c
e-b-X-Y-e-b
…

a) The AIG of user X and item Y

b) A transformed representation of the associative interaction graph

c) Sample random walks

Figure 5.2 The Associative Interaction Graph of A User-item Pair

After the random walk paths are generated, I calculate their probability of existence. At a certain time, a random walk may jump from one node to its neighbors or stop following a probability distribution. Thus, the random walk path $h = n_1 \rightarrow n_2 \rightarrow \cdots \rightarrow n_l$ will have a probability of existence of $P(h \mid G) = p_t(n_2 \mid n_1)$

$p_t(n_3 \mid n_2) \cdots p_t(n_l \mid n_{l-1}) p_s(n_l)$, where p_t is the transit probability and p_s is the stop probability.

Next, I define similarities between random walk paths $K_h(h_i, h_j)$ as the product of the similarities of matching nodes and links on the paths. Since I focus on one user-item pair for each AIG, I always match the focal user-item pairs when comparing the random walk paths. If two random walk paths do not have a one-to-one (node and link) mapping after I match their focal user-item pairs, I simply deem their similarity as 0, otherwise I define it as: $K_h(h_i, h_j) = K_{node}(n_0^{h_i}, n_0^{h_j}) \times K_{link}(n_0^{h_i} \to n_1^{h_i}, n_0^{h_j} \to n_1^{h_j}) \times K_{node}(n_1^{h_i}, n_1^{h_j}) \times \cdots \times K_{node}(X^{h_i}, X^{h_j}) \times K_{node}(Y^{h_i}, Y^{h_j}) \times \cdots \times K_{node}(n_{l-1}^{h_i}, n_{l-1}^{h_j}) \times K_{link}(n_{l-1}^{h_i} \to n_l^{h_i}, n_{l-1}^{h_j} \to n_l^{h_j}) \times K_{node}(n_l^{h_i}, n_l^{h_j})$, where $n_t^{h_i}$ is the tth node on the random walk path h_i, $K_{node}()$ is the kernel representation of node features, and $K_{link}()$ is the kernel representation of link features.

Finally, I define the graph kernel as the sum of pairwise similarities of random walk paths weighted by their existence probability: $K(G_x, G_y) = \sum \sum \left(K_h(h_i, h_j) \times P(h_i \mid G) \times P(h_j \mid G) \right)$. In this research, I normalize the graph kernel for better prediction performance.

In my design, the graph kernel is essentially a convolution kernel. It meets the semi-positive definite property required by kernel methods. The graph kernel covers both the structure and the node/link features of the AIG. It is able to accumulate node/link features by following the graph structure to focal user-item pairs.

5.4 Experimental Study

5.4.1 Dataset

In this research, I used three datasets to evaluate the performance of the proposed approach. 1) An online book retail dataset obtained from a major Chinese online bookstore in Taiwan. The dataset includes 3 years of transactions of 2,000 randomly selected users, involving 9,695 books and 18,771 transactions. 2) An online clothing retail dataset provided by a leading U.S. online clothing merchant. The dataset includes 16 million online transactions from a 3-month period, involving 4 million households and 128,000 items. 3) A book rating dataset collected from the Book-Crossing community (Ziegler et al., 2005). The dataset reports 278,858 users' 1,149,780 ratings on 271,379 books. In this research I treated a rating as a transaction and ignored the rating grade.

Table 5.2 Dataset Statistics

Dataset	Number of users	Number of items	Number of transactions	Avg. purchases per user	Avg. sales per item
Book Retail	851 (~2,000)	8,566 (~9,700)	13,902 (~18,000)	16.34 (~9)	1.62 (~1.86)
Clothing Retail	1,000 (~4 million)	7,328 (~128,000)	9,332 (~16 million)	9.33 (~4)	1.27 (~125)
Book Rating	1,000 (~280,000)	15,578 (~270,000)	19,329 (~1.15 million)	19.33 (~4.12)	1.24 (~4.24)

* The numbers in parentheses are the statistics on the original dataset.

For the experiments, I included only the users with 5 to 100 transactions for meaningful testing. This range constraint resulted in 851 users for the book retail dataset. For comparison purposes, I sampled 1,000 users with 5 to 100 items from the clothing retail dataset and book rating dataset. Table 5.2 reports the descriptive statics of the three reduced datasets. The reduced data set were split into 80% training data and 20% testing data according to transaction time. The items that were only in the testing data, which cannot be predicted by the algorithm, were removed.

5.4.2 Experimental Procedures

To evaluate the effectiveness of the proposed method, I conducted two sets of experiments. Experiment I compares the proposed algorithm with the state-of the art heuristic algorithms that use graph structure. For a fair comparison, only transaction information is used in this set of experiments. Experiment II compares the proposed algorithm with learning-based algorithms that use local features or limited graph-related features. For this experiment, available user/item information is used.

In experiment I, I compared the performance of the proposed approach with five major popular recommendation algorithms: 1) a user-based algorithm that cross-recommends items to users who have interacted with similar items; 2) an item-based algorithm that cross-recommends items whose users are similar; 3) an item popularity algorithm that recommends items according to its sales/access volumes; 4) a spreading activation algorithm (Huang et al., 2004b), which is a PageRank-like node ranking algorithm; and, 5) a link analysis algorithm (Huang et al., 2007b), which is a HITS-like node ranking algorithm. In this set of experiments, I calculated the graph kernel based on the training data instances and trained a one-class SVM classifier. The classifier was applied onto all user-item pairs without links. For each user, the items with the highest rank according to the prediction confidence were considered as recommendations. For the benchmark algorithms, similar procedures were conducted.

In experiment II, I compared the performance of the proposed approach with two learning-based algorithms that use no or limited graph information: 1) the set kernel, which is a convolution kernel that considers each of the nodes on the AIGs equally and sums pairwise node comparisons of the nodes on the AIGs for focal user-item pairs' similarity: $K_{set}(G_x, G_y) = \sum_{i \in G_x} \sum_{j \in G_y} K_{node}(i, j)$; 2) The local feature-based method, that considers only the local features of the focal user-item pair. The similarity between two user-item pairs is calculated as the product of their user similarity and the item similarity.

In experiment II, all the models can utilize local features on users and items. For the three datasets in the experiments, different local features can be extracted. 1) For the book retail dataset, user information includes "year of birth" and "education level;" item information includes "book title," "keywords," and "introduction." 2) For the clothing retail dataset, user information is not available; item information includes "product category" and "description." 3) For the book rating dataset, user information includes "location" and "age;" item information includes "book author." I applied different kernels to represent these local features to a kernel form. For categorical data, such as "location," "book author," "category," "description," etc., I applied a linear kernel to represent them in a kernel form. For numerical data, such as "age" and "education level," I applied a Radial Basis Function (RBF) kernel to represent them in a kernel form. For textual data, such as "book title," "keywords," etc., I applied a linear kernel with a bag-of-words model to represent them in a kernel form.

In experiment II, the proposed graph kernel, the set kernel, and the kernel on local features were implemented based on the split training and testing data. A one-class SVM classifier was trained for customer information and one classifier was trained for item information for each algorithm. The classifiers were applied onto the user-item pairs without links. For each user, the highly ranked items according to the prediction confidence were considered to be recommendations.

5.4.3 Evaluation Metrics

For evaluation, I generated a ranked recommendation list of N items for each user and compared the recommendations with actual transactions in the testing data. I adopted three types of evaluation metrics:

1) The precision, recall, and F-measure of the top-10 recommendations (Pazzani, 1999).

$$\text{Precision}=\frac{\text{Number of recommended products that match with the future purchases}}{\text{Total number of recommended products}}$$

$$\text{Recall}=\frac{\text{Number of recommended products that match with the future purchases}}{\text{Total number of products with future purchases}}$$

F-measure= 2*Precision*Recall/(Precision+Recall)

2) The rank score that emphasizes the first couple of recommendations (Pazzani, 1999).

Rank score $R=100\frac{\sum_i R_i}{\sum_i R_i^{\max}}$, where $R_i=\sum_j \frac{p(i,j)}{2^{(j-1)/(h-1)}}$ and

$$p(i,j)=\begin{cases}1, & \text{if product } j \text{ is in customer } i\text{'s future purchase list}\\ 0, & \text{otherwise.}\end{cases}.$$

3) The ROC curve that represents the global performance of recommendations.

5.5 Results and Discussion

5.5.1 Graph Kernel vs. Heuristics

Table 5.3 Top 10 Recommendation Performance for Graph-based Algorithms

Dataset	Algorithms	Precision	Recall	F-measure	Rank Score
Book Retail	User-based	0.0242	0.1165	0.0377	8.2882
	Item-based	0.0076	0.0396	0.0121	2.4338
	Item popularity	0.0258	0.1317	0.0405	**12.4843**
	Link analysis	**0.0280**	**0.1408**	**0.0439**	**11.8745**
	Spreading activation	0.0224	0.1110	0.0349	9.3618
	Graph kernel	**0.0286**	**0.1461**	**0.0449**	11.2838
Clothing Retail	User-based	0.0114	0.0778	0.0193	**4.5900**
	Item-based	0.0078	0.0597	0.0136	3.0354
	Item popularity	0.0062	0.0326	0.0100	1.4173
	Link analysis	**0.0124**	**0.0818**	**0.0209**	**4.7752**
	Spreading activation	0.0076	0.0513	0.0129	3.1005
	Graph kernel	**0.0131**	**0.0855**	**0.0219**	**4.1307**
Book Rating	User-based	**0.0082**	**0.0269**	**0.0119**	**1.8260**
	Item-based	0.0052	**0.0203**	0.0079	**1.3080**
	Item popularity	0.0058	**0.0224**	0.0089	**1.9283**
	Link analysis	0.0065	**0.0225**	0.0096	**1.6260**
	Spreading activation	**0.0071**	**0.0264**	**0.0106**	**1.7307**
	Graph kernel	**0.0077**	**0.0295**	**0.0118**	**1.6251**

* Boldfaced measures are not significantly different from the largest measure at the 10% significance level

Table 5.3 reports the performance of top 10 recommendations by the models that use the graph structure. In general, according to precision, recall, and F-measure for the top-10 recommendations, the proposed graph kernel is always in the group of best algorithms. The other heuristic algorithms' ranks vary on different datasets, which is because their models were not learned from the data. According to the rank score measure, which values the first couple of recommendations more than later recommendations, the proposed algorithm is slightly lower than the link analysis algorithm or the item popularity algorithm in different datasets, although the performance difference is minor.

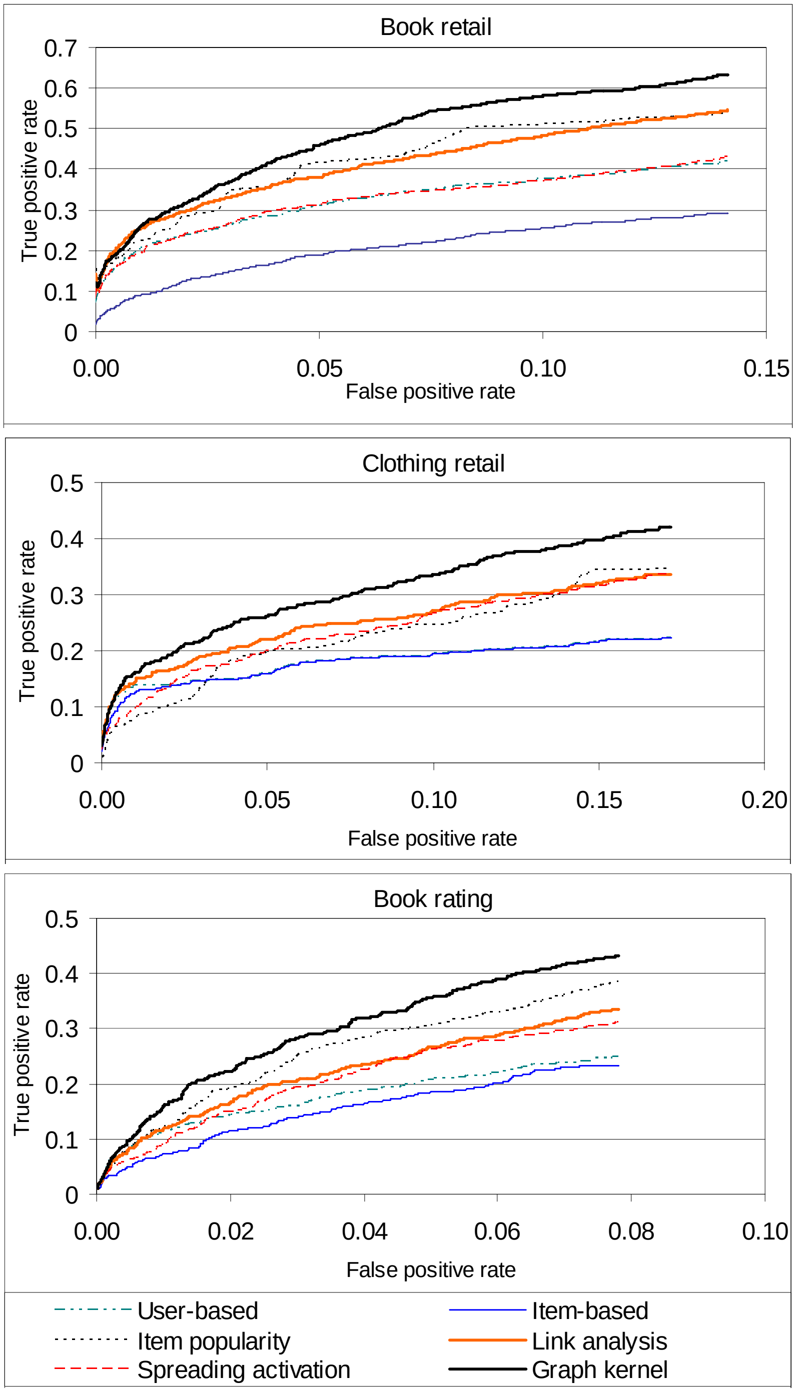


Figure 5.3 ROC Curves of Top 1,000 Recommendations of Graph-based Algorithms

Figure 5.3 reports the ROC curves for the top 1,000 recommendations made by the six different models. It shows that when a small number of predictions were needed, the graph kernel was among one of the best algorithms. When a large number of predictions were needed, the graph kernel significantly outperformed all other methods. The graph kernel targets at link prediction on the entire graph. It may provide more "high confidence" predictions to some users than others. In a top-N recommendation task, many of these predictions may not be considered and used. Enlarging the number of recommendations takes other "high confidence" predictions into consideration. In real life applications, it represents those cases to recommend items to frequent users. The proposed algorithm will provide more benefits in such circumstances.

5.5.2 Graph Kernel vs. Other Learning-based Algorithms

Table 5.4 reports the performance of top 10 recommendations by different learning-based algorithms. In most of the experiments, the graph kernel outperformed both the set kernel and method on local features in precision, recall, F-measure, and rank score. In the book rating dataset, if item information was used, the local feature-based method had the best precision, recall, and F-measure.

Figure 5.4 reports the ROC curves for the top 1,000 recommendations made by the learning-based algorithms. In most of the experiments, the graph kernel significantly outperformed the set kernel and local features. When using item information on the book rating dataset, the local feature method archived the best performance at first, which was outperformed by the graph kernel later. In this dataset, the item information was author name. In the book rating application, the author information had a strong correlation with users' ratings. A reader may choose a book purely because of the author. On the other hand, books rated by a reader may have minor topic/style similarities. Compared with the author information, co-rated books (the AIGs) may have significantly less information that can interoperate with readers' choice.

Table 5.4 Top 10 Recommendation Performance for Learning-based Algorithms

Dataset	Node information	Algorithm	Precision	Recall	F-measure	Rank score
Book retail	User information	Graph kernel	**0.0267**	**0.1354**	**0.0418**	**12.0860**
		Set kernel	0.0053	0.0292	0.0086	0.8428
		Local feature	0.0102	0.0514	0.0160	3.0465
	Item information	Graph kernel	**0.0262**	**0.1338**	**0.0412**	**12.4667**
		Set kernel	0.0074	0.0362	0.0114	1.0491
		Local feature	0.0007	0.0026	0.0011	0.1277
Clothing retail	Item information	Graph kernel	**0.0080**	**0.0463**	**0.0132**	**2.9329**
		Set kernel	0.0046	0.0351	0.0080	1.0879
		Local feature	0.0006	0.0037	0.0001	0.3487
Book rating	User information	Graph kernel	**0.0054**	**0.0235**	**0.0085**	**1.6750**
		Set kernel	0.0017	0.0073	0.0026	0.3351
		Local feature	0.0017	0.0062	0.0026	0.6212
	Item information	Graph kernel	0.0071	0.0289	0.0108	**2.0688**
		Set kernel	0.0010	0.0044	0.0016	0.7701
		Local feature	**0.0102**	**0.0507**	**0.0158**	**1.7072**

*Boldfaced measures are not significantly different from the largest measure at the 10% significance level.

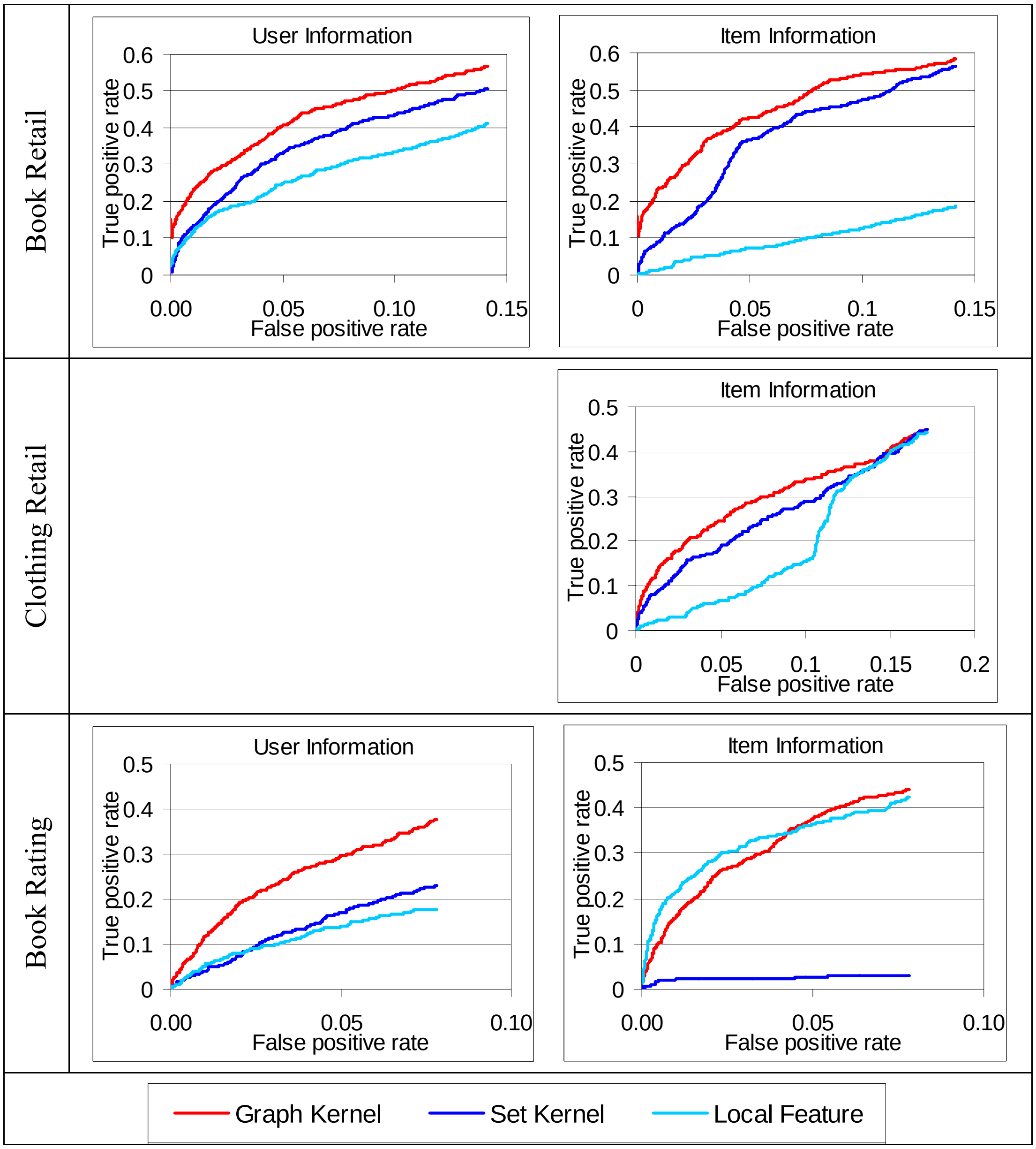


Figure 5.4 ROC Curves of Top 1,000 Recommendations of Learning-based Algorithms

Compared to the set kernel and the local feature method, the graph kernel makes better use of the user-item graph, which may cause its better performances. In the graph kernel the features of individual nodes can be considered to be propagated to the focal user-item pairs following the links. The set kernel considers only the nodes in the AIG without utilizing the network structure. The local feature-based method even does not use the neighbor’s node information. These design deficiencies may have limited their prediction abilities.

5.6 Summary

In this chapter I presented a graph kernel-based approach for making recommendations. Treating the recommendation task as a link prediction problem in user-item interaction graph, I defined an associative interaction graph for each user-item pair and use the structure of the graph to infer whether or not the user-item pair may have a link. I proposed a graph kernel that can effectively capture graph-related features from the AIGs. In the experiments on three real-world datasets, the proposed method achieved nearly the best precision, recall, and F-measure for the top-10 recommendations as compared to graph-based heuristics and learning-based algorithms that use less graph information. Moreover, if a larger number of recommendations are needed, the proposed method achieves significantly better performance than all the benchmark algorithms.

Using the application in recommender systems, this chapter shows the effectiveness of using graph structure in the link prediction task. With the explorations in the previous three chapters, it is shown that graph structure is correlated with node information (in node classification) and link information (in link prediction). In the next chapter, I will study the community detection task, in which I examine whether the node information or link information could help detect multiple entities with close interactions (i.e., dense local graph structures). I will examine the problem in the context of online interactions in the next chapter.

CHAPTER 6. COMMUNITY DETECTION: EXPLORING LINK SENTIMENTS TO DETECT ONLINE SOCIAL GROUPS

6.1 Introduction

In previous chapters, I studied the node classification task and the link prediction task. In this chapter, I move on to study the relationships among multiple nodes and deal with the community detection task. I explore the use of link sentiment information to identify groups of individuals who have similar interests and activities according to their online communications.

With the development of the Internet and computer mediated communication (CMC) techniques, humans' social interaction patterns have been significantly changed. In addition to traditional social activities, which were limited by the spatial factor, people now extend their friendships from neighborhoods to the Internet (Wellman, 2005). The advance of Web 2.0 provides several new communication channels, such as Web blogs, online review Websites, Web forums, social networking Websites, etc., for individuals to express their opinions and interact with each other. With these interactive social media, individuals can form online virtual social groups. Table 6.1 provides some examples of the communication channels and the social groups they may lead to.

Online interactions usually involve a large number of participants. Grouping persons with similar interests and activities into social groups or communities could make it easier to assess online opinions and to trace the opinion leaders and followers. The community detection problem in social networks has attracted several researchers' interest. Most previous studies deemed social relationships as binary relations, in which the links only indicate the existence of the relationship. Such a simplified model cannot capture the rich information in social networks, especially the sentiment (favorable vs. unfavorable, agreement vs. disagreement) of the interactions. Such sentiments may be more important in online interactions, in which people show stronger sentiments than in face-to-face interactions (Sia et al., 2002).

Table 6.1 Web 2.0 Techniques and Possible Social Groups

Web 2.0 media	Examples	Contents	Possible social groups
Web blog	Myspace.com	Diverse opinions from the blogger	Opinion leaders and followers
Online review Website	Eopinions.com, Youtube.com, Digg.com	Reviews targeted on selected items	People with similar interests on targeted items
Web forum	Walmartblows.com	Discussions related to selected themes	People hold similar opinions on selected topics
Social networking Website	Secondlife.com, Facebook.com	Casual talks	Friendship

In this research, I propose to label links in social networks with the sentiment of social relations (positive/negative) and construct sentiment social networks for community detection. I propose a framework to extract sentiment social networks from online communications and a GN-H co-training algorithm that can use both positive and negative sentiments in SSN for community detection. I evaluate the performance of using sentiment information in community detection with simulated data and conduct a case study on an online review dataset (www.eopinions.com) to show the utility of the proposed framework.

This chapter is organized as follows. Section 6.2 reviews previous community detection studies. Section 6.3 introduces the proposed community detection framework on sentiment social networks. Section 6.4 describes the experiments on simulated data and an online review dataset. Section 6.5 discusses the results. Section 6.6 summarizes the findings.

6.2 Literature Review

Social relations and social networks in organizations have been considered to have an affect on the adoption of IT artifacts (Bruque et al., 2008; Montazemi et al., 2008) and knowledge diffusion (Oh et al., 2005; Robert et al., 2008). At the same time, social interactions are being changed by information technology and IT artifacts. For example, groupware has changed people's collaboration patterns in decision making (Huang and Wei, 2000; Gemino et al., 2005). Internet and computer mediated communication have changed our daily communication patterns (Watson-Manheim and Belanger, 2007).

The recent development of Web 2.0 media has led to a great amount of online interactions (Jones et al., 2004) and computerized communities from traditional neighborhoods (Wellman, 2005). Different types of participants form online communities with different characteristics (Gu and Konana, 2007) and contribute knowledge to the online community (Bieber et al., 2002; Ma and Agarwal, 2007). The community factors in online interactions also affect real-world business problems (Chua et al., 2007a). Accessing the community structure of online interactions can help us study online opinions and their impact on business applications.

6.2.1 Community Detection

Previous literature argues that community studies should be approached more from a network analytic perspective (Piselli, 2007). The network of social interactions embedded in online communications can be used to detect groups of participants with similar interests and activities. At the algorithm level, such a community detection problem in social networks is similar to the graph partition problem in computer science (Karypis and Kumar, 1998) and the gene clustering problem in medical informatics (D'haeseleer et al., 2000) . In this research, I propose a taxonomy to review previous community detection literature. While focusing on the applications in social networks, I also include other application areas due to their algorithm innovations.

As shown in Figure 6.1, previous community detection studies can be characterized by network types, network characteristics, and algorithm types.

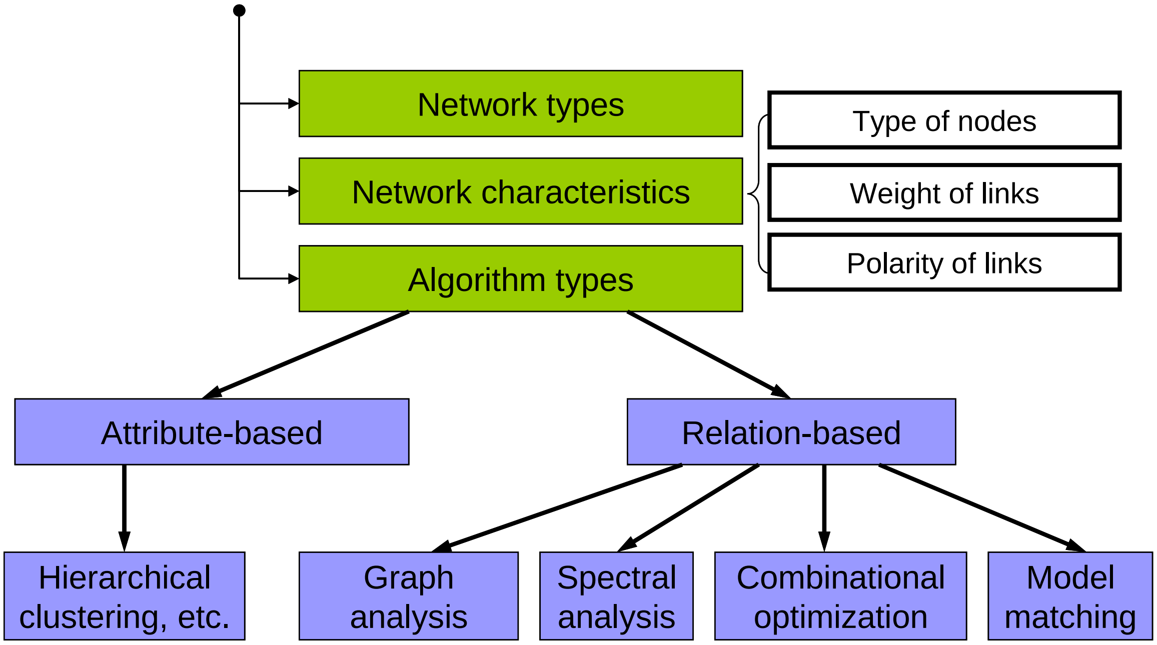


Figure 6.1 A Taxonomy for Community Detection in Network Analysis

6.2.1.1 Network Types

Previous community detection studies have been conducted on various types of networks, including both social networks and non-social networks. The examined social networks included co-authorship networks (Girvan and Newman, 2002), friendship networks (Airoldi et al., 2008), email networks (Guimera et al., 2006), and purchase relationship (between buyers and sellers) networks (Reichardt and Bornholdt, 2007). Community detection studies have also been conducted on Webpage networks (Flake et al., 2002), paper citation networks (Hopcroft et al., 2004), biological networks (e.g., gene interaction network) (Wilkinson and Huberman, 2004), zoological networks (e.g., food web) (Newman, 2004a), linguistics networks (e.g., word semantic network) (Capocci et al., 2005), financial networks (e.g., stock price correlation network) (Barber, 2007), and product networks (e.g., product co-sales network) (Clauset et al., 2004).

Most studies on social networks are constructed on the basis of certain communication channels which can be classified into two types, face-to-face interactions and online interactions. Social networks based on face-to-face interactions include traditional co-authorship networks and real-world friendship networks. Examples of social networks based on online interactions include email networks, online friendship networks (e.g., friendships in Web forums), and purchase relationship networks in e-commerce Websites.

6.2.1.2 Network Characteristics

The networks analyzed in previous community detection studies can be characterized by their node types, weight of links, and polarity of links.

According to the types of nodes, networks can be classified as unipartite networks, which contain only one type of nodes, bipartite networks, which contain two types of nodes, and multi-partite networks, which contain multiple types of nodes. A social network is usually studied as a unipartite network in which the nodes are persons. Multi-partite networks have been used to model the word semantic network, which is a type of linguistics networks (Newman and Leicht, 2007). The nodes in the network could be nouns, adjectives, or other types of words (Newman and Leicht, 2007).

Networks can be classified as unweighted networks and weighted networks according to the weight of the links. In social networks, the weight of links can be used to represent the strength of social relations, such as the strength of friendships and frequency of email contacts. However, most previous studies on social networks ignore such weight information and use the links only to represent the existences of relationships.

Polarity of links can be considered as a special type of link weight, i.e., whether the link is positive or negative. In social networks, link polarity can be used to represent sentiment of social interactions, for example, whether two persons like each other or hate each other or whether two persons agree with each other or not. According to this measure, networks can be classified as polar networks and non-polar networks, where a polar network may contain both positive and negative links and non-polar networks only contain positive (or negative) links.

6.2.1.3 Algorithm Types

Previous community detection algorithms can be classified into attribute-based algorithms and relation-based algorithms. An attribute-based algorithm takes advantage of only node attributes or views links as some special node attributes. Under such a design most traditional clustering algorithms can be directly applied on the community detection problem. For example, after defining a similarity measure based on node attributes, such as Pearson correlation measure, both hierarchical clustering (Girvan and Newman, 2002) and K-means (Hopcroft et al., 2004) have been applied to group similar nodes together. Since the links are converted to node features in the attribute-based algorithms, they can be directly applied to weighted networks and polar networks. If a proper node similarity function is designed, they can also be applied to bipartite and multi-partite networks. Although attribute-based algorithms make limited use of relational information in the data, they are easy to extend to complicated networks.

Relation-based algorithms make use of relational information in the network for community detection. They can be further categorized into graph analysis methods, spectral analysis methods, combinational optimization methods, and model matching methods.

1) Graph analysis methods

Graph analysis methods take a network view of the data in community detection. For example, fully connected nodes (cliques) can be used to build K-clique chains, i.e., a chain of connected K-cliques, as indicators of communities (Palla et al., 2005; Farkas et al., 2007).

One major graph analysis method is divisive algorithms that remove links from networks to generate isolated components as communities. For example, the max-flow-min-cut method in graph theory has been applied to split Webpage networks into clusters (Flake et al., 2002). As a milestone algorithm, the GN algorithm removes the high betweenness links to generate communities (Girvan and Newman, 2002). The same researchers proposed a modularity measure (Newman and Girvan, 2004) to determine the best community generated from the algorithm. The GN algorithm has been extended to weighted networks (Newman, 2004a). In addition, Monte Carlo estimation has been applied to approximate betweenness measures in the algorithm to improve its computational efficiency (Wilkinson and Huberman, 2004). Using design methodology similar to Girvan and Newman's, other studies proposed to gradually remove the links that cause the biggest change to the average shortest path length (Fortunato et al., 2004) and the links with the highest edge clustering coefficient (Radicchi et al., 2004) to detect communities.

Another type of graph analysis method is random walk-based algorithms that model networks with random walks to pursue the community structure. Among these algorithms, the Markov Cluster Algorithm (MCL) is one successful method that has been widely used in bioinformatics (Enright et al., 2002). The algorithm first "expands" each node's links to its indirect neighbors, then "inflates" the weights of the links to remove less important links. Zhou proposed node distance measures based on random walks to cluster "closer" nodes together (Zhou, 2003a, 2003b). Alves designed clustering algorithms based on random walk transit probabilities between nodes and put node pairs with higher transit probabilities into the same communities (Alves, 2007). The model is able to model both unweighted and weighted networks. Raghavan et al. proposed a label propagation method in which each node gradually adopts the majority its neighbors' labels (Raghavan et al., 2007). The algorithm is efficient and can address the community detection problem on very large-scale networks.

2) Spectral analysis methods

Spectral analysis methods take a matrix view of the network. Based on the spectral graph theory, the eigenvectors of representative matrices of graphs are related to the graphs' community structures. In previous computer science research, spectral analysis on the Laplacian matrix has been widely adopted in addressing the graph partition problem (Donetti and Munoz, 2004). Capocci et al. applied spectral analysis methods on a normal matrix derived from the adjacency matrix for community detection on word semantic networks (Capocci et al., 2005). Newman applied the spectral analysis method on the modularity matrix, which is derived from their proposed modularity measure, for community detection (Newman, 2006b, 2006a). Barber generalized the modularity measure to bipartite networks in the framework of spectral analysis methods (Barber, 2007).

3) Combinational optimization methods

Combinational optimization methods take a search approach for community detection. Assuming each node can be assigned into one of the N clusters, they look for a combination of the nodes (in different clusters) that can provide a better community structure. After defining the quality of community assignments as an objective function, all traditional combinational optimization algorithms can be directly applied to find a community assignment that can maximize the objective function.

The modularity measure (Newman and Girvan, 2004) is a well-adopted quality measure for combinational optimization methods. Greedy search method (Clauset et al., 2004; Newman, 2004b), simulated annealing (Guimera and Amaral, 2005; Reichardt and Bornholdt, 2007) and extremal optimization (Duch and Arenas, 2005) have been applied on this measure to find a better community. Recent research found that the modularity measure has a “resolution limit” that can only detect coarse-level partitions (Fortunato and Barthelemy, 2007). Thus, Arenas et al. modified the modularity measure to address this problem (Arenas et al., 2008a; Arenas et al., 2008b) by applying extremal optimization and tabu search. Rosvall and Bergstrom proposed another quality measure based on the mutual information of the network’s original topology and community level topology (Rosvall and Bergstrom, 2007). They applied the simulated annealing algorithm to maximize this measure for community detection.

4) Model matching methods

Model matching methods take a probabilistic view of the networks. They aim to design probabilistic models on node distributions (in different clusters) and link distributions (inter and intra-clusters) and find appropriate parameters of the model that can generate networks that are similar to the original network. Hastings assumed that there is a uniform probability for inter-cluster links and a uniform probability for intra-cluster links, and used the belief propagation algorithm to find the node assignments that have the highest probability of generating a given network (Hastings, 2006). Newman and Leicht assumed that nodes in a cluster have similar distributions to link to other nodes and find out the analytical solutions that maximize the likelihood of generating the network given the model (Newman and Leicht, 2007). Airoldi et al. proposed a membership stochastic model that captured both cluster-level inter-connection patterns and node-level characteristics of the connections for the generation of networks (Airoldi et al., 2008). The model parameters were estimated based on a mean-field variational inference method and an E-M algorithm.

6.2.2 Research Gaps

Table 6.2 Previous Studies on Community Detection

Studies	Network Types	Network Characteristics			Algorithm Types
		Node*	Weight#	Polarity+	
(Enright et al., 2002)	Biological network	U	U	N	Graph analysis
(Flake et al., 2002)	Webpage network	U	U	N	Graph analysis
(Girvan and Newman, 2002)	(F2F) Co-authorship network; Zoological network	U	U	N	Graph analysis
(Zhou, 2003a)	(F2F) Friendship network; (F2F) Co-authorship network; etc.	U	U	N	Graph analysis
(Zhou, 2003b)	Biological network	U	U	N	Graph analysis
(Clauset et al., 2004)	Product network	U	U	N	Combinational optimization
(Donetti and Munoz, 2004)	(F2F) Friendship network; (F2F) Co-authorship network	U	U	N	Spectral analysis
(Fortunato et al., 2004)	(F2F) Friendship network; Zoological network	U	U	N	Graph analysis
(Hopcroft et al., 2004)	Paper Citation Network	U	U	N	K-Means
(Newman, 2004a)	Zoological network; Linguistics network	U	W	N	Graph analysis
(Newman, 2004b)	(F2F) Co-authorship network	U	U	N	Combinational optimization
(Radicchi et al., 2004)	(F2F) Co-authorship network	U	U	N	Graph analysis
(Wilkinson and Huberman, 2004)	Biological network	U	U	N	Graph analysis
(Capocci et al., 2005)	Linguistics network	U	W	N	Spectral analysis
(Duch and	(F2F) Friendship network;	U	U	N	Combinational

Arenas, 2005)	(OL) Email network; (F2F) Co-authorship network; etc.				optimization
(Guimera and Amaral, 2005)	Biological network	U	U	N	Combinational optimization
(Palla et al., 2005)	(F2F) Co-authorship network; Biological network	U	U	N	Graph analysis
(Guimera et al., 2006)	(OL) Email network	U	U	N	Graph analysis
(Hastings, 2006)	N/A	U	U	N	Model matching
(Newman, 2006a)	(F2F) Friendship network; (F2F) Co-authorship network; etc.	U	U	N	Spectral analysis
(Newman, 2006b)	(F2F) Friendship network; (F2F) Co-authorship network; etc.	U	U	N	Spectral analysis
(Alves, 2007)	(F2F) Friendship network	U	W	N	Graph analysis
(Barber, 2007)	(F2F) Friendship network; Financial network	B	U	N	Spectral analysis
(Farkas et al., 2007)	(F2F) Co-authorship network; Stock network	U	W	N	Graph analysis
(Newman and Leicht, 2007)	(F2F) Friendship network; Word network	U	U	N	Model matching
(Palla et al., 2007)	(F2F) Co-authorship network; (OL) Email network	U	U	N	Graph analysis
(Raghavan et al., 2007)	Biological network; (F2F) Friendship network	U	U	N	Graph analysis
(Reichardt and Bornholdt, 2007)	(OL) Purchase network	U	U	N	Combinational optimization
(Rosvall and Bergstrom, 2007)	(F2F) Friendship network; etc.	U	U	N	Model matching
(Arenas et al., 2008a)	(F2F) Friendship network	M	U	N	Combinational optimization
(Arenas et al., 2008b)	(F2F) Friendship network	U	U	N	Combinational optimization
(Airoldi et al., 2008)	(F2F) Friendship network; Biological network	U	U	N	Model matching

* Node: U - unipartite; B - bipartite; M - Multi-partite;
# Weight: U - unweighted; W - weighted;
+ Polarity: N – non-polar; P - polar.

Table 6.2 summarizes previous studies on community detection. One may notice that social networks were the major application domain of previous research. In addition, most of these studies on social networks were based on face-to-face interactions (annotated by F2F in Table 6.2) collected from survey or co-authorship data. Only a few studies were based on online interactions (annotated by OL in Table 6.2). As one type of Web 2.0 online interaction, the seller-buyer networks extracted from e-commerce transactions were studied in (Reichardt and Bornholdt, 2007).

In previous research, most studies were conducted on unipartite, unweighted, and non-polar networks. From a social network perspective, that means the links were indicators of the existence of certain relationships between individuals. Few studies considered sentiments (polarity) of the links, which is common in social interactions.

The literature review shows that most research took a relation-based approach. One reason is that the attribute-based algorithms have been well-studied in traditional clustering research. The other reason could be the better performance that can be achieved by using relational information (Girvan and Newman, 2002). Among the four types of relation-based algorithms, graph analysis algorithms were more frequently used. This type of algorithms has also been applied to weighted networks in previous studies.

Noticing the limited research on community detection in online interactions and on using link polarity (sentiment) in community detection, I focus on the following two research questions in this research: 1) How can we effectively detect communities from online interactions? 2) Will using link polarities (sentiments) improve community detection effectiveness?

6.3 Research Design

6.3.1 Research Framework

In this research, I propose to extract social networks from online interactions and apply community detection algorithms to find individuals with more consistent interest and activities. In addition, I propose to take the sentiments embedded in online interactions into consideration. Figure 6.2 shows my framework for extracting sentiment social networks from online interactions and detecting communities from the networks. The framework contains four steps: data collection, sentiment social network construction, community detection, and evaluation.

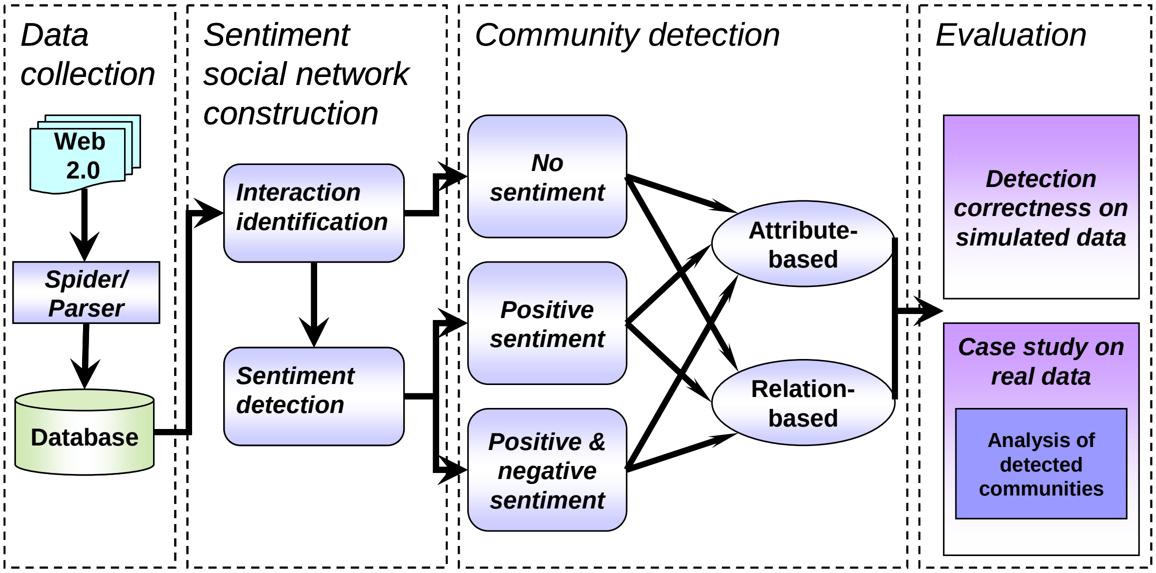


Figure 6.2 A Framework for Community Detection from Online Interactions

Online interactions are usually documented in comments and postings on Web 2.0 Websites, such as Web forums, Web blogs, etc. In the data collection stage, such data need to be collected using Web spiders or retrieved through Web APIs. The extracted data is usually in free-text format, which needs to be parsed into relational databases for further analysis.

At the sentiment social network construction stage, I first extract social interactions from the communications and then annotate them with interaction sentiments. In Web forums, detecting social interactions (i.e., the reply relationship between postings) is non-trivial, since many postings do not specify this information. Thus, text mining approaches can be applied to extract such relationships (Fu et al., 2008). In online product/news review applications, the reviewed items can be considered a proxy of online social interactions, where everyone's reviews respond to others' reviews on the same item. In sentiment social networks, "sentiment" may have subtle differences in meaning in different applications. In friendship networks, it may mean favorable and unfavorable sentiments. In online interactions, it means the agreement of individual opinions, which may be related to favorable and unfavorable sentiments. To annotate two opinions' or arguments' agreement, previous research extracted word features and sentiment from the messages and applied machine learning algorithms to assess their agreement (Hahn et al., 2006; Stavrianou and Chauchat, 2008). In online product/news reviews, the sentiment (agreement) of interactions can be measured by the differences between people's ratings on the same products. In this research, I consider sentiment social networks at the person level and aggregated all agreement/disagreement of online opinions as indirectional links between node pairs.

At the community detection stage, I apply both attribute-based and relation-based algorithms on the sentiment social networks. To answer the research questions, I derive three types of networks from each original sentiment social network: one network without sentiment information, one network with only positive sentiment information, and one network with both positive and negative sentiment information. The two sets of algorithms are applied onto the three networks to examine the effect of using sentiment information in community detection.

At the evaluation stage, the community detection algorithms' performances can be measured if the networks' community structure is known, such as on simulated data. In addition to such strict evaluations, I conduct a case study on a real-world dataset to show the utility of community detection in helping us understand the opinions and opinion leaders in online interactions.

6.3.2 Algorithm Design

In this research, I choose hierarchical clustering as the attribute-based algorithm. The algorithm can directly handle weighted links and links with positive or negative sentiments. I choose the GN algorithm (Girvan and Newman, 2002) as the relation-based algorithm, since it is one of the most successful community detection algorithms in previous literature. The GN algorithm can be applied in weighted networks. However it cannot handle negative links at all. Thus, I design a GN-H co-training algorithm as the relation-based algorithm on networks with both positive and negative sentiments. Since the two sets of algorithms used in this research can both generate a series of community assignments, I adopt and modify the modularity measure to select the best one for output.

6.3.2.1 Modularity Measure

The modularity measure was originally proposed by Newman and Girvan (Newman and Girvan, 2004) and has been widely used to select better community assignments from a series of candidates. The measure is defined as:

$$\mathrm{Q}=\frac{1}{2\mathrm{m}}\sum_{ij}\left(A_{ij}-\frac{k_i k_j}{2m}\right)\delta\left(c_i,c_j\right)$$

where m is the total number of edges (or the sum of edge weights); A_{ij} is the weight (sentiment) of the link between node *i* and *j*; and k_i (k_j) is the degree (or the sum of edge weights) of node *i* (*j*). $\delta(c_i,c_j)=1$ if $c_i=c_j$, i.e., *i* and *j* are in the same community. It is designed to favor intra-cluster links rather than inter-cluster links. The community assignments with more intra-cluster links and less inter-cluster links have a higher modularity measure and are considered a better community assignment.

The original modularity measure can be applied only to networks without negative sentiments. Thus, I modify it for community detection in networks with both positive and negative sentiments. I first split the sentiment social network into a network with positive sentiments and a network with negative sentiments.

The modified measure is then defined as:

$$\mathrm{Q}=\frac{1}{2\mathrm{m}^{+}+2m^{-}}\left[\sum_{ij}\left(A_{ij}^{+}-\frac{k_i^{+}k_j^{+}}{2m^{+}}\right)\delta\left(c_i,c_j\right)-\sum_{ij}\left(A_{ij}^{-}-\frac{k_i^{-}k_j^{-}}{2m^{-}}\right)\delta\left(c_i,c_j\right)\right]$$

where m^{+} (m^{-}) is the total number of edges (or the sum of absolute edge weights) of the positive (negative) sentiment network; A^{+}_{ij} (A_{ij}^{-}) is the absolute weight (sentiment) of the link between node *i* and *j* in the positive (negative) sentiment network; and k^{+}_{i} and k^{+}_{j} (k_i^{-}

and k_j^-) are the degrees (or the sum of absolute edge weights) of node *i* and *j* in the positive (negative) sentiment network. $\delta(c_i,c_j)=1$ if $c_i=c_j$, i.e., *i* and *j* are in the same community. The modified measure can be considered as combining two original measures applied onto the positive sentiment social network and negative sentiment social network, respectively. The community assignments with more intra-cluster positive links (and less inter-cluster positive links) and less intra-cluster negative links (more inter-cluster negative links) have higher modularity measures and are considered better assignments.

After modifying the modularity measure, combinational optimization algorithms can be directly applied to detect communities in sentiment social networks. However, in this research I adopt and develop graph analysis methods as examples of relation-based algorithms, due to their good performance and ease of interpretation.

6.3.2.2 Hierarchical Clustering

For the attribute-based algorithm, I apply the hierarchical clustering algorithm (Johnson, 1967) which can be applied on networks with both positive and negative sentiments. In this algorithm, each node's neighbors (links) are considered as its features. The link weight and sentiment (if it exists) are used as feature values. For two nodes *i* and *j* with feature vectors V_i and V_j, I define their similarity using the Pearson correlation efficient measure: $C_{ij}=\frac{\sum_k (V_{ik}-\mu_i)(V_{jk}-\mu_j)}{n\sigma_i\sigma_j}$, where μ_i (μ_j) is the mean of the vector V_i (V_j); σ_i (σ_j) is the standard deviation of the vector V_i (V_j); and *n* is the length of the feature vectors, i.e., the number of nodes in the network. After defining the similarity measure, the single link hierarchical clustering algorithm is applied to group nodes together. The single link hierarchical clustering algorithm starts from considering each node as a cluster. It then combines clusters that have the most similar nodes. The algorithm generates a series of communities, among which the best one can be selected according to the (modified) modularity measure.

6.3.2.3 The GN Algorithm

For the relation-based algorithm on networks without negative sentiment information, I apply the GN algorithm, which is an effective graph analysis algorithm in community detection. It gradually removes the high betweenness links from the graph to split the graph into isolated components, which are considered as detected communities in the network. The algorithm can generate a series of communities, among which the best one can be selected according to the modularity measure.

6.3.2.4 The GN-H Co-training Algorithm

For the relation-based algorithm on networks with both positive and negative sentiments, I propose a GN-H co-training algorithm which combines the GN algorithm with hierarchical clustering to detect communities.

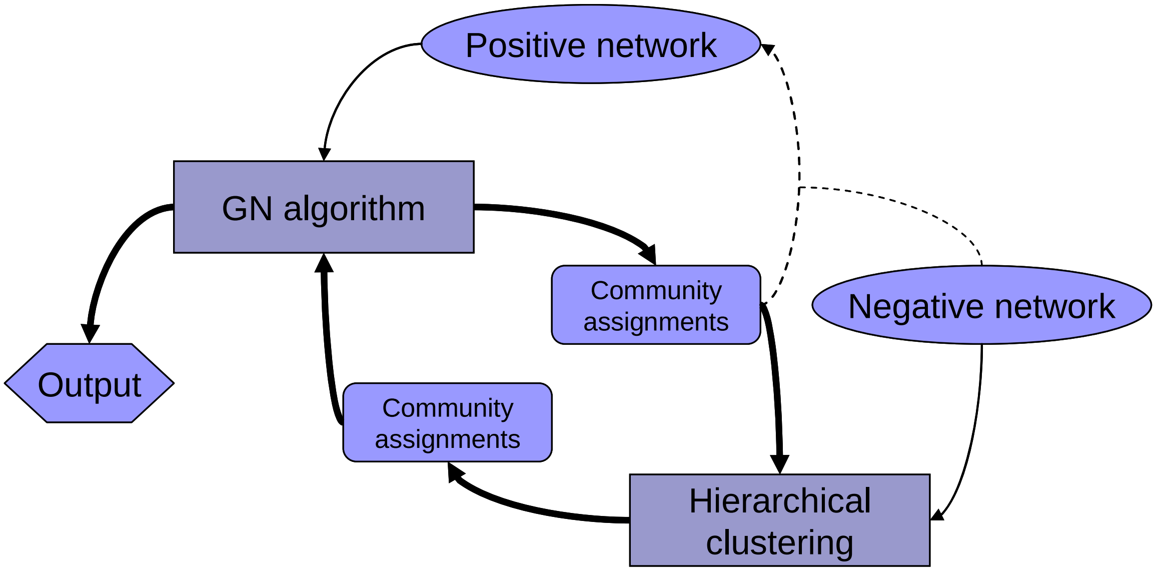


Figure 6.3 The GN-H Algorithm

As shown in Figure 6.3, in the GN-H algorithm the original sentiment social network is split into a positive network and a negative network. The GN algorithm is applied on the positive part, while hierarchical clustering is applied to the negative part. In this process, the community detection results of the GN algorithm are incorporated into hierarchical clustering and the community detection results of hierarchical clustering are incorporated into the GN algorithm. In hierarchical clustering the nodes that have similar (negative) link distributions to the clusters generated by the GN algorithm are considered as having a higher probability to be in the same community. In the GN algorithm, the shortest paths between the nodes in a cluster generated by hierarchical clustering are weighted less than original paths, so that they will have smaller probability to be separated to different communities. In addition, I also generate hidden positive links based on their similarities in negative link distributions and combine them with original positive links to be used in the GN algorithms. For multiple iterations of the algorithm, the generated communities from the GN-part with the highest modularity measure are considered as the entire algorithm’s output (see details in Figure 6.4).

```
1) Split the inputted sentiment social network $G$ to a positive part $G^+$ and a negative part $G^-$.
2) Using $G^+$ as the input of GN algorithm $G_{GN}$, generate communities $C_{GN}$.
3) Represent each node $i$ as a feature vector $NV_i$ based on its negative links in $G^-$ to each of the communities in $C_{GN}$.
   $NV_{ij} = \frac{\text{sum of the absolute weight of links from node i to cluster j}}{\text{number of nodes in cluster j}}$.
4) Define node similarity $S(i,j)$ as the inner product of their feature vectors $<NV_i,NV_j>$.
5) Apply hierarchal clustering based on $S(i,j)$ and generate communities $C_h$.
6) Select top $m$ links according to $S(i,j)$, which do not belong to $G^+$. Combine the $m$ links and $G^+$ as the input of GN algorithm $G_{GN}$.
7) Apply the GN algorithm based on both $G_{GN}$ and $C_h$ to generate output $C_{GN}$. In the GN algorithm, the shortest paths between the nodes in a same community of $C_h$ are weighted as half of the original paths when calculating links’ betweenness.
8) Go to step 3 and iterate multiple times. For the multiple iterations, keep the community assignment $C_{GN}$ generated by GN-part with the highest modularity measure as the entire algorithm’s output.
```

Figure 6.4 Pseudo Code of the GN-H Algorithm

6.4 Experimental Study

In previous community detection research, the effectiveness of proposed algorithms was usually evaluated on simulated data with known community structure. I adopted this approach to examine the effect of using sentiments in community detection. In addition, I conducted a case study on a product review dataset to exemplify the utility of the proposed framework in helping us understand online opinions and activities. In these experiments, I only considered the link polarity and ignore the weight of links to focus on the effect of link sentiments.

6.4.1 Datasets

1) Simulated data

I adopted a widely used data generation methodology for simulated data (Girvan and Newman, 2002; Radicchi et al., 2004; Guimera and Amaral, 2005). The generated network contains 128 nodes that belong to 4 clusters (32 nodes each). The average degree for each node is 16 (i.e., in total 1,024 links). The links were generated randomly, which contains α portion of positive links and 1-α portion of negative links. Both positive links and negative links can be inter-community and intra-community links. Among the negative links there were β portion intra-cluster links and 1-β portion inter-cluster links. For the positive links there were γ portion intra-cluster links and 1-γ portion inter-cluster links. Since positive links indicate people agreeing with each other and negative links indicate people disagreeing with each other, a smaller β and a larger γ indicates that the network shows clearer community structure, while vice versa means the network has less clear community structure. I generated networks for α from 0.1 to 0.9 (step 0.1), β from 0 to 0.2 (step 0.05), and γ from 0.5 to 0.9 (step 0.1). (Since the network has only four clusters, β should be less than 0.25 and γ should be larger than 0.25. I considered β≤0.2 and γ≥0.5 as more realistic parameter settings that may happen in real data.) The combination of the three parameters led to 225 parameter settings. For each parameter setting, I randomly generated 100 networks to test different algorithms.

2) Real-world dataset

In this research, I used an online product review dataset spidered from www.eopinions.com (http://www.trustlet.org/wiki/Downloaded_Epinions_dataset) (Massa and Avesani, 2007) to show the utility of the proposed framework. This dataset contains 49,290 users who rated 139,738 items in 664,824 reviews. The ratings are from 1 to 5 where 1 means the best and 5 means the worst. To control the experiment size and focus on more active users, I reduced the eopinions dataset by randomly choosing 10% of the users who have more than 5 product reviews on file. If two users had co-reviewed a product, they were considered having an online interaction, where the absolute differences of their ratings were used to represent interaction sentiments. For each pair of users, I aggregated their interactions and average the sentiment values on all links between them. If two users had only co-reviewed one or two products, the link was considered unreliable and discarded. If a user pair's average sentiment value was less than 2, they were considered as having a positive interaction. If such value was larger than 2 they were considered as having a negative interaction. The final network contains 1,037 users, 6,808 positive links, and 505 negative links.

6.4.2 Experiment Procedure

To compare the effect of using sentiment in community detection on simulated data, I generate three networks for each simulated network. The original simulated network contains both positive and negative sentiments. The network with positive sentiment was generated by removing all negative links. The network without sentiment information was generated by considering all links equally. The attribute-based and relation-based algorithms were applied on the networks. The results were compared with the actual cluster when generating the data for evaluation.

The similar three types of networks with different sentiment information were derived from the network extracted from the eopinions dataset,. I only apply relation-based algorithms, which are the focus of this research, to generate the clusters on these three networks. The statistics of the generated communities were examined for the utility of the proposed framework.

6.4.3 Evaluation Metrics

For the simulated data, the underlying community structure is known. I adopted the normalized mutual information measure (Danon et al., 2005), which was widely used in evaluating clustering algorithms, to evaluate community detection algorithms' performances. To calculate this measure, one needs to construct an $n*m$ confusion matrix N, where n is the number of real communities, m is the number of generated communities, and N_{ij} is the number of nodes in the real community i that has been assigned to community j by the algorithm. The normalized mutual information measure is

$$Normalized\ M.I. = \frac{-2\sum_{i=1}^{n}\sum_{j=1}^{m} N_{ij}\log\left(\frac{N_{ij}N_{..}}{N_{i.}N_{.j}}\right)}{\sum_{i=1}^{n} N_{i.}\log\left(\frac{N_{i.}}{N_{..}}\right) + \sum_{j=1}^{m} N_{.j}\log\left(\frac{N_{.j}}{N_{..}}\right)},$$

where $N_i.$ is the sum of row *i*; $N_{.j}$ is the sum of column *j*; and N.. is the sum of all elements. A higher value of the normalized mutual information measure indicates better community detection results.

For the real-world dataset, the underlying community structure is unknown. Thus, I inspected the mesoscopic description of the network (i.e., community-level topology structure of the network) resulting from the relation-based algorithms. I also studied the opinion leaders and key opinions for the major communities identified to assess the utility of conducting community detection analysis on online interactions.

6.4.4 Hypotheses

In correspondence with the research questions, I tested two hypotheses in the experiments on the simulated data:

H1. Differentiating link sentiments (using positive sentiments) outperforms not differentiating link sentiments (not using sentiments) in community detection.

H2. Using both positive and negative sentiment information outperforms using only positive sentiment information in community detection.

6.5 Results and Discussion

6.5.1 Experiments on Simulated Data

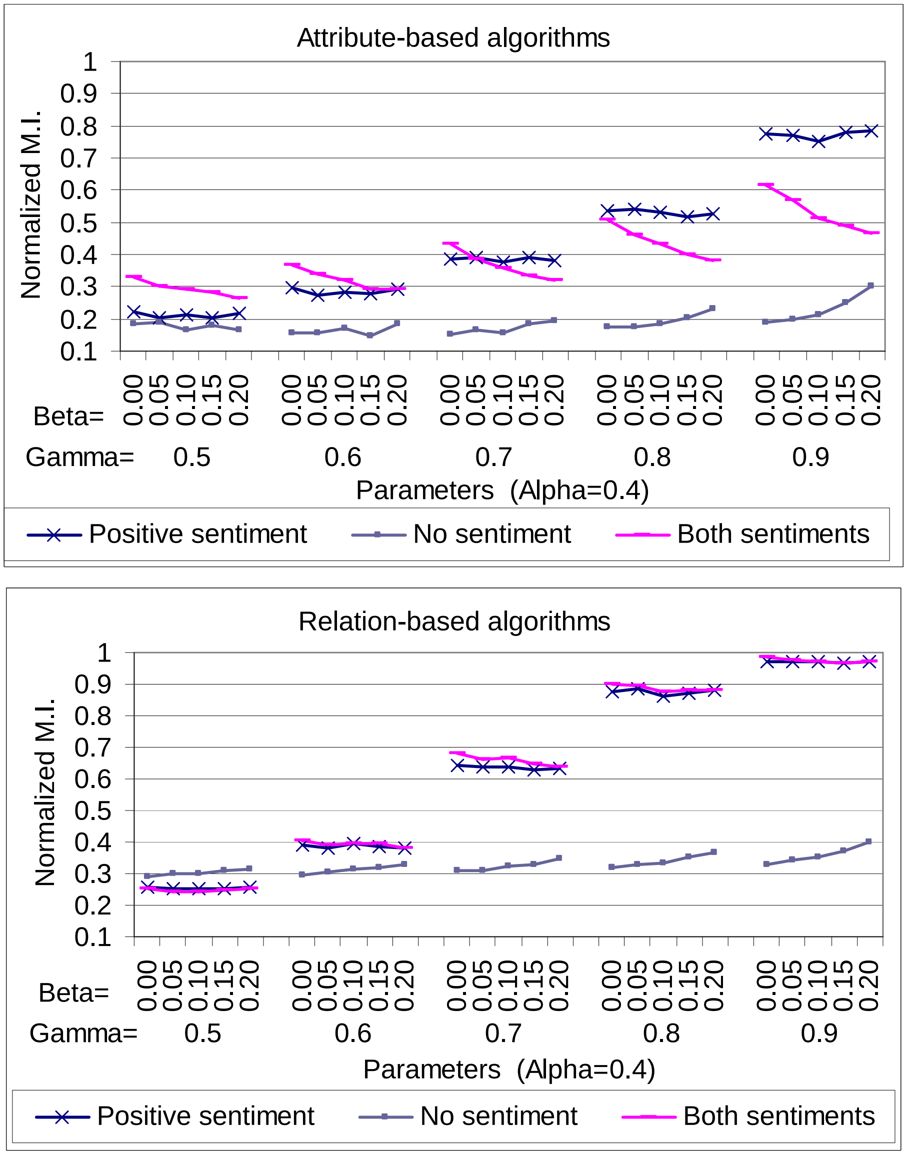


Figure 6.5 Community Detection Performance on Simulated Data when α=0.7

Figure 6.5 exemplifies the attribute-based and relation-based algorithms' performances on simulated data when α =0.7. The algorithms' performances on other parameter settings show similar trends. On the graph, it is clear that including sentiment information significantly outperformed excluding sentiment information in community detection, whether attribute-based or relation-based algorithms were used. In addition, the proposed GN-H algorithm that uses both types of sentiments outperformed the GN algorithm that uses only positive sentiments in most parameter settings. For the attribute-based algorithm, i.e., hierarchical clustering, the use of both types of sentiments outperformed the use of positive sentiments only when γ is small, i.e., positive links are less correlated with the community structure. The hierarchical clustering algorithm does not effectively use sentiment information. The effect of negative sentiments only shows when the positive sentiments are not useful enough.

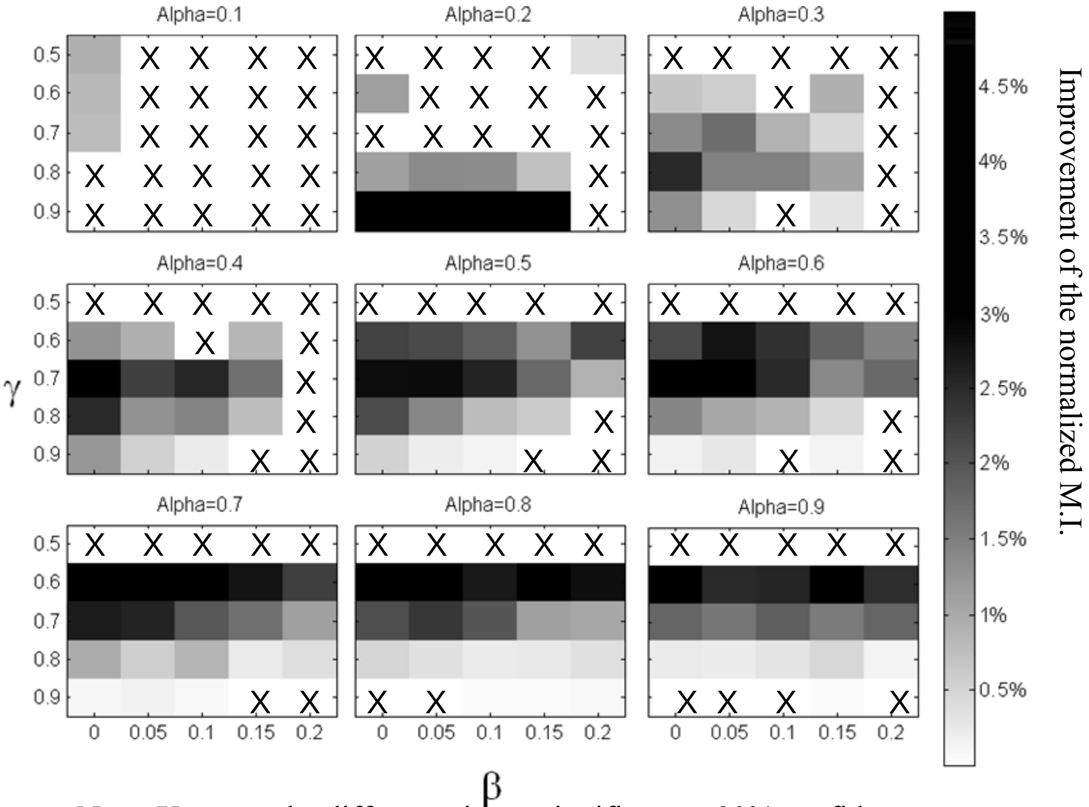
Table 6.3 Hypotheses Testing

H1 pairwise t-test: positive sentiment >no sentiment		
Number of parameter settings	GN algorithm	Hierarchal clustering
Hypothesis confirmed at 99% confidence interval	186	212
Hypothesis confirmed at 95%~99% confidence interval	0	6
Hypothesis confirmed at 90%~95% confidence interval	2	1
Hypothesis rejected at 90% confidence interval	37	6

H2 pairwise t-test: both sentiments > positive sentiment		
Number of parameter settings	GN/GN-H algorithm	Hierarchal clustering
Hypothesis confirmed at 99% confidence interval	104	66
Hypothesis confirmed at 95%~99% confidence interval	12	7
Hypothesis confirmed at 90%~95% confidence interval	11	2
Hypothesis rejected at 90% confidence interval	98	150

Table 6.3 shows the pairwise t-test results on the total 225 parameter settings for the two hypotheses. It is clear that H1 was supported on most of the parameter settings. For H2, the hypothesis was partially supported. The hierarchical clustering algorithm

cannot effectively use negative sentiment information and the hypothesis was rejected for 2/3 of the parameter settings for this algorithm.



Note: X means the difference is not significant at 90% confidence interval in pairwise t-tests.

Figure 6.6 Performance Improvement of GN-H Algorithm over the GN Algorithm

To better show the hypothesis testing results on H2 for the GN-H algorithm, I visualize its result in Figure 6.6, where a darker cell shows the higher performance improvement of the GN-H algorithm (using both positive and negative sentiments) over the GN algorithm (using positive sentiment). X means the improvement is not significant at the 90% confidence interval in pairwise t-tests. In general, the GN-H co-training algorithm significantly outperformed the GN algorithm on most parameter settings when α is larger than 0.2 and γ is larger than 0.5. In other words, for the networks have more positive links and the positive links indicate clearer community structure, the hypothesis H2 is confirmed for the GN-H algorithm.

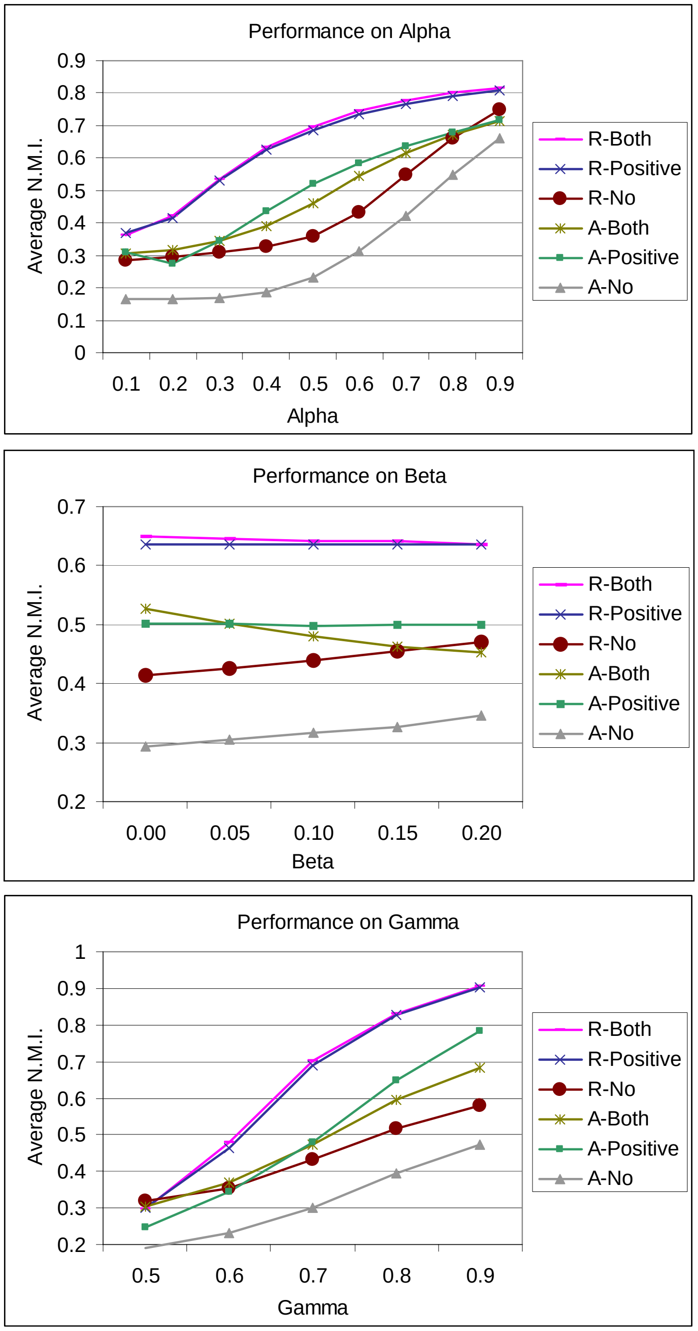


Figure 6.7 Aggregated Performance on α, β, and γ

Figure 6.7 reports the aggregate performances of all algorithms according to the three parameters that control the generation of the simulated data. In general, the proposed GN-H algorithm outperformed all other algorithms on the aggregated measures. In addition, I noticed that all algorithms' performances are positively correlated with α and γ. This means that positive links play important roles in community detection in sentiment social networks. When the network contains more positive links and the positive links indicate clearer community structure, the algorithms' performances improve. I also noticed that the performances of the algorithms that use negative sentiments have a negative correlation with β. Since a smaller β indicates that negative links show clear community structure, this phenomenon clearly shows the effect of using negative sentiments in community detection. Furthermore, the performances of the algorithms that deem negative links as positive links have a positive correlation with β, which indicates that these algorithms use negative sentiments reversely (or wrongly). In community detection, it is critical to differentiate positive and negative sentiments.

6.5.2 Experiments on the Eopinions Dataset

I applied three relation-based algorithms (GN algorithm on network without sentiments, GN algorithm on network with positive sentiment, and GN-H algorithm with both sentiments) on the eopinions dataset. As shown in Figure 6.8, the communities generated by all three algorithms show clearer structure as compared to the original node level representation. The three algorithms' results have similar structures and all have two major communities and some smaller communities with close interconnections.

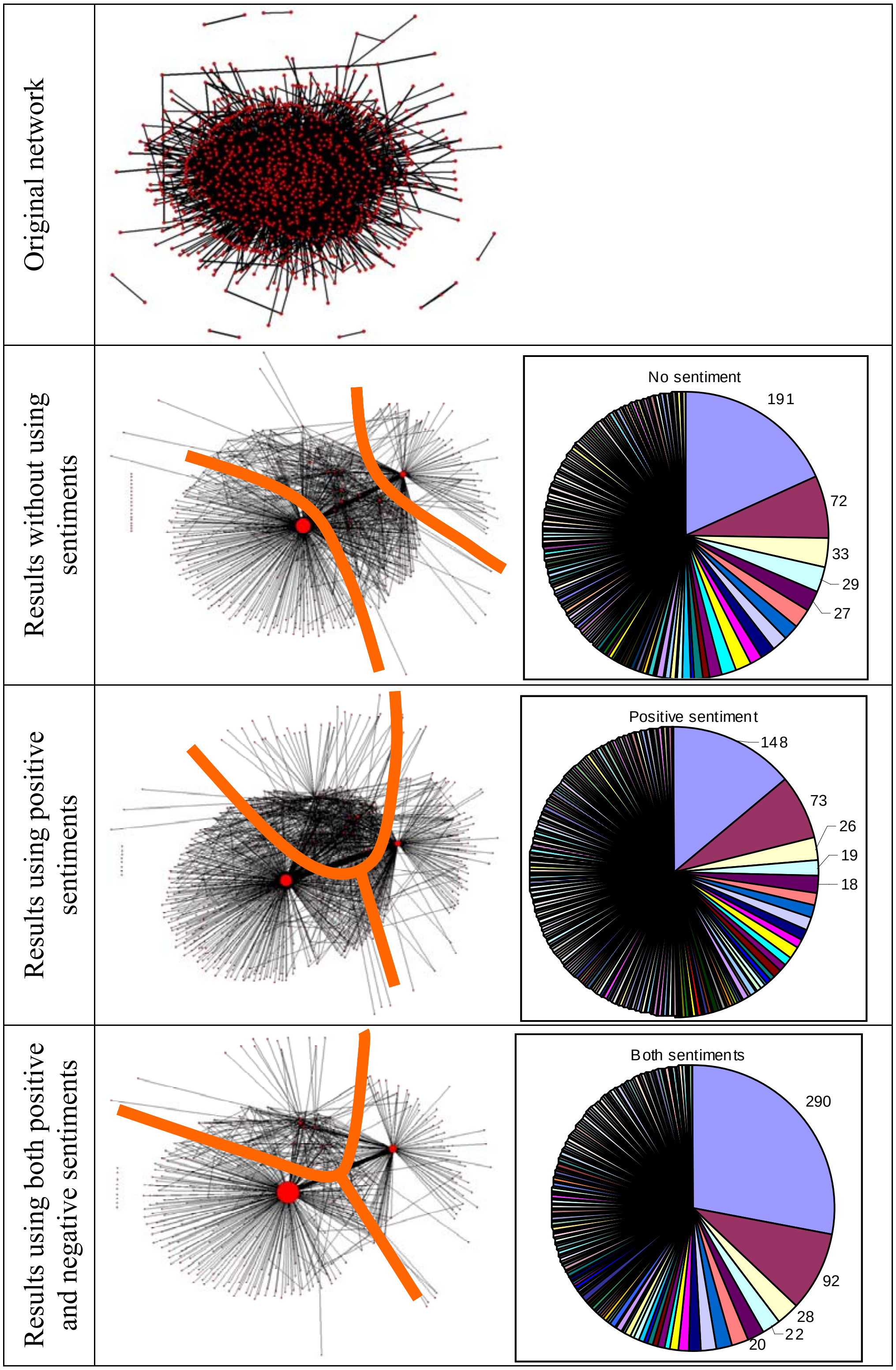


Figure 6.8 Community Detection Results on the Eopinions Dataset

The GN algorithm that used no sentiment information generated 438 communities; its intra-cluster negative link ratio is 6.63% and inter-cluster negative link ratio is 7.16%, which are similar to the overall negative link ratio of the network (6.91%). The GN on the positive sentiment network has 5.00% intra-cluster negative link ratio and 8.25% inter-cluster negative link ratio. The GN-H algorithm using both sentiments has 5.75% intra-cluster negative link ratio and 8.41% inter-cluster negative link ratio. Both of them were able to put less negative links inside communities and put more negative links between communities as compared to the one that used no sentiment information. According to the generated community structure, the GN-H algorithm using both sentiments and the GN algorithm that use only positive sentiments are more similar to each other.

Table 6.4 Characteristics of the Two Major Communities

		Largest community	Second largest community
Negative link ratio	No sentiment	6.93%	6.13%
	Positive sentiment	6.14%	3.43%
	Both sentiments	6.61%	3.57%
Different algorithms' community overlap	No sentiment vs. positive sentiment	141	59
	Positive sentiment vs. both sentiments	141	61
	Both sentiments vs. no sentiment	179	56
	All three algorithms	137	64

Table 6.4 reports some characteristics of the two major communities identified by the three algorithms, respectively. Obviously, major portions of the communities generated by different algorithms are the same. However, the algorithms using sentiment information provide communities with smaller intra-cluster negative link ratios,

especially for the second largest community. Such structural differences were caused by the small portions of non-overlapped nodes, which show the effect of considering sentiment information in community detection.

Table 6.5 Top 10 High-degree Users in the Two Major Communities

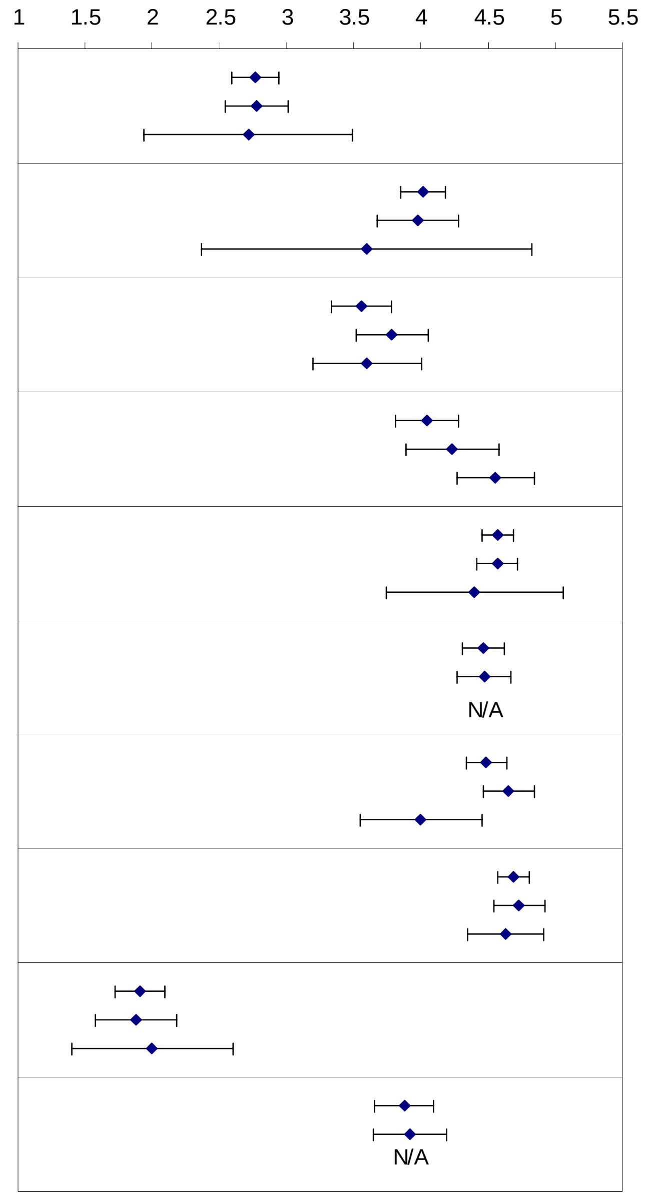
Top 10 users	No sentiment			Positive sentiment			Both sentiments		
	User ID	Intra-cluster degree	Overall ranking	User ID	Intra-cluster degree	Overall ranking	User ID	Intra-cluster degree	Overall ranking
Largest community	4855	99	5	3037	89	8	5734	139	1
	3037	98	8	4855	86	5	7585	113	4
	5304	96	7	7585	85	4	3037	112	8
	7585	94	4	5304	83	7	4855	111	5
	3604	88	11	3604	80	11	3604	110	11
	6334	80	13	6334	77	13	5304	109	7
	12724	78	20	17165	67	15	2051	98	12
	17165	76	15	12983	63	14	17165	89	15
	8584	74	21	8584	63	21	6334	88	13
	28164	72	16	12780	61	31	12983	85	14
Second largest community	776	51	10	776	50	10	2397	48	25
	2397	45	25	2397	50	25	768	44	35
	310	38	55	310	41	55	310	44	55
	3796	38	17	14428	41	30	3796	43	17
	14428	37	30	3796	41	17	14428	43	30
	768	36	35	768	39	35	1138	34	46
	1211	30	82	517	33	9	302	33	42
	302	30	42	302	33	42	2565	33	22
	1138	29	46	1138	32	46	34581	33	38
	10069	29	85	1985	31	59	1985	32	59

Table 6.5 reports the top 10 opinion leaders of the two major communities according to their degree (i.e., number of peoples who agree with them) in the community. It also reports their overall ranking according to their degree in the entire network. In general, the three algorithms generated similar top 10 opinion leaders for the two major communities. It is noticed that these opinion leaders of major communities

were necessary high-degree users in the network. The opinion leaders of the largest community usually rank around 5-15 on the entire network, while the opinion leaders of the second largest community usually rank around 20-50. Several other high-degree users on the entire network may be users who posted a lot of reviews and were involved in a lot of online interactions but did not have a consistent like/dislike pattern. Such users will not have a high impact on the other users in the community, since they may not be considered as friends/peers by other users. The high-degree users in individual communities, on the other hand, have higher impact on a community's behavior. They deserve further analysis and long term observation.

Figure 6.9 shows the two major communities' ratings on the most frequently rated items together with their global ratings, using the communities detected by the GN-H algorithm as an example. While many items' ratings are consistent for different communities, there is some evidence of the different interests of individual communities. For example, the second largest community's average rating on item 660 is significantly higher than the global rating at the 90% confidence interval. The second largest community's average rating on item 6114 is significantly lower than that of the largest community. In addition, the numbers of ratings on the items are not balanced, which also indicates individual communities' different interests. For example, the second largest community's users do not like to rate items 11436 and 734. But they rated item 1083 quite often. Such phenomena can also be found on the results provided by the other two algorithms. In general, the identified communities may have significantly different opinions. To assess online opinions, it is necessary to inspect discussions at the community level.

Group name	Community*	Number of ratings
363	A	173
	L	102
	S	7
659	A	132
	L	44
	S	5
390	A	123
	L	79
	S	5
660	A	91
	L	34
	S	9
615	A	89
	L	49
	S	5
11436	A	82
	L	51
	S	0
6114	A	80
	L	29
	S	12
1083	A	80
	L	30
	S	19
7611	A	78
	L	32
	S	10
734	A	73
	L	49
	S	1



* A: overall rating; L: rating of the largest community; S: rating of the second largest community.
# The bars show 90% confidence interval.

Figure 6.9 Different Communities' Interests

6.6 Summary

In this research, I proposed to account for social interaction sentiments in tackling the community detection task. I proposed a framework to extract sentiment social networks from online interactions and proposed a GN-H co-training algorithm that can handle both positive and negative sentiment information for community detection. Experiments on simulated data showed that differentiating sentiments would significantly benefit the community detection task. In addition, using both positive and negative sentiments in the GN-H algorithm could further improve community performance. Experiments on a product review dataset also showed that considering sentiment information led to community assignments with fewer negative links inside of communities and more negative links between communities. Analysis on the generated communities suggested that opinion leaders and opinions extracted from individual communities could provide us with more detailed information of online community behaviors. It is worthwhile to analyze online interactions at the community level.

This chapter explored the community detection task using an example application of online communications. Together with the previous chapters, it shows the necessity of combining node/link information with graph structure in graph-based learning problems. I will further explicate the findings, contributions, and implications of the graph-based learning framework in the next chapter. I will discuss its relevance to Management Information Systems research and possible extensions in the future.

CHAPTER 7. CONCLUSIONS AND FUTURE DIRECTIONS

7.1 Contributions

The work presented in this dissertation highlights several methods and applications in graph-based learning. It has made several theoretical and practical contributions which can be found useful to future researchers.

In Chapter 2 I presented two examples that showcase the effective application of network typology analysis methods in improving our understanding about the global characteristics of relational information extracted from documents. The first study in Chapter 2 analyzed gene interaction networks extracted from biomedical literature using various information extraction methods. The second study in the chapter analyzed the patent citation networks aggregated to different analytical unit levels. While the analysis in this chapter provides us with some concrete results on the two types of networks' structural characteristics, a major contribution of this study is to show the advantage of using the network topology analysis to compare relational information extracted from the same data sources using different extraction methods. In studies of information systems, the extracted relations represent the outputs of various IT artifacts and have their semantic implications. The global characteristics of these outputted relations help us better understand the IT artifacts. Moreover, this chapter proposed a general framework to conduct network topology analysis on relational information extracted from literature, which can be used in other research in knowledge mapping, knowledge diffusion, and information extraction.

Chapter 3 addressed the patent classification problem as a node classification task. As a major theoretical contribution of this chapter, I proposed incorporating knowledge evolution processes in addressing knowledge management tasks. I used citation networks to represent the knowledge evolution processes in patents and designed different kernel functions under a kernel-based framework to capture the structures of the patent citation networks. I found that features of cited patents and the structure of patent citation networks both play important roles in patent classification. I also found that combining information in citation networks with patent contents benefited the classification. This chapter also made a critical practical contribution to patent management. In the experiments, the proposed method achieved more than 30 percent performance improvement in accuracy as compared with the state-of-the-art algorithms. It showed the potential to reduce human effort in patent pre-classification in patent offices' daily operations. This study also lent support to a policy that requires inventors to file patent citations together with their patent applications.

In Chapter 4, I addressed a gene function prediction problem using gene interaction networks. From a theoretical perspective this chapter is an extension of the previous chapter. This chapter used a context graph concept to represent the related features in neighbor nodes to facilitate focal nodes' classification. I proposed a context graph kernel to capture such features to address the gene function prediction problem. Moreover, I proved the matrix formulation and the convergence characteristics of the context graph kernel. All these theoretical findings can be extended to other node classification problems as long as a context assumption can hold. In addition, as a particular contribution to the medical informatics domain, the improved prediction performance of the proposed method as compared to other state-of-the-art methods offers biologists a more sophisticated IT artifact to support their research.

In Chapter 5, I addressed the recommendation system problem as a link prediction problem in user-item interaction networks. I defined an associative interaction graph for each user-item pair and captured its structure with a graph model to infer the possibility of the existence of the link. This chapter continued making theoretical contributions through taking advantage of the graph structure of the data. Furthermore, I extended the kernel-based framework and introduced one-class classification for this task. From the perspective of link prediction, this extension is necessary and would help improve the performance of link prediction. This research can be directly applied to e-commerce applications to improve customer online shopping experiences.

Chapter 6 focused on the community detection problem in online environments. This research made multiple contributions to research in online communities. First of all, I identified sentiment as an important link attribute that affects the effectiveness of community detection. This finding may prompt future research to combine sentiment analysis with social network analysis into a next stage of sentiment social network analysis. Second, I proposed a general framework that can be used to detect communities in online communications. The framework can be applied in various contexts with light adaption, such as Web forums, Web blogs, social Websites, etc. Finally, the proposed GN-H co-training algorithm provided an effective routine for designing sentiment-based communication detection algorithms. I expect more design science research will be conducted in this direction in the future.

7.2 Relevance to Management Information Systems Research

Knowledge discovery is critical to organizations' knowledge creation and knowledge management. As an IT artifact studied by information systems (IS) researchers for years, knowledge discovery modules have been embedded in various information systems and affected daily businesses activities (Matheus et al., 1993). Knowledge discovery from graph-structured data is relevant to IS research not only from the methodological perspective, but also from a practical perspective. Graph-structured data widely exist in intra-organizational social relations, computer mediated communications, and the technology adoption process. These application areas have been of the interest to IS researchers for years. The graph-based learning framework proposed in this dissertation provides more methods that support these studies.

The research in the information systems discipline has been classified into behavioral science and design science paradigms (Hevner et al., 2004). Design science studies targets producing technology-based IT artifacts, such as constructs, models, methods, or instantiations, to relevant business problems. The studies conducted in this dissertation follow the design science paradigm and are aimed at creating a set of new and innovative methods under the graph-based learning framework to reduce the gap between growing amounts of (graph-structured) data and human beings' cognition limitations in acquiring knowledge from the data for decision making. In this dissertation, I have applied the graph-based learning framework in order to address several critical business-related problems, such as the patent classification problem in knowledge management, the recommender system problem, and the community detection problem in online communications. The framework has also been applied to medical informatics applications. The essays in this dissertation exemplify the design process of several artifacts to address the four graph-based learning tasks. In this graph-based learning framework, the identified knowledge discovery tasks, proposed knowledge discovery approaches, and examined knowledge discovery applications in this dissertation contribute to our knowledge in information systems and can be adopted by future researchers in new applications.

7.3 Future Directions

Although this dissertation has addressed several challenges in graph-based learning, future research will continue to broaden and deepen our understanding in knowledge discovery on graph-structured data from the following directions:

1) Explore the use of more complicated graph information in knowledge discovery. In this dissertation, most examined networks are unipartite networks (except Chapter 5), most of the networks contain only one type of links (except Chapter 6), and all the applications are conducted on a single network. While these applications perfectly showcased the proposed graph-based learning framework, there could be additional applications with more complicated graph-structured data relevant to the real-world. In the future, I will explore knowledge discovery problems in multi-partite networks, with multiple types of links, or by combining multiple networks under the proposed graph-based learning framework.

2) Explore more effective methods for the four graph-based learning tasks. While this dissertation has addressed some aspects of the four tasks, these problems are far from being solved from the methodological perspective. For future business applications and information systems research, it is necessary to explore more effective, efficient, and scalable techniques.

3) Explore applications that can be better addressed by the framework. In business intelligence, computer mediated communication, knowledge management, etc., there are a large number of problems that involve graph-structured data and that can be better addressed using a graph-based learning framework. I plan to further expand the application area of the proposed framework in the future.

APPENDIX A: MATRIX FORMULATION OF THE CONTEXT GRAPH KERNEL

Proposition:

In a genome's gene interaction network G with n genes $\{g_1, g_2, \ldots g_n\}$, the context graphs of gene g_x is represented as G_x. If $\tilde{K} = \{\tilde{K}_{i,j}\} = \{K(G_i, G_j)\}$ is the context graph kernel matrix of the entire genome; $M=\{M_{i,j}\}=\{p_t(g_j|g_i)\}$ is the transition probability matrix; $Q=\{Q_{i,j}\}=\{p_s(g_i)\}$ is the stopping probability matrix; and $K_0 = \{K_{0_{i,j}}\} = \{K_g(g_i, g_j)\}$ is the node information kernel matrix. The context graph kernel matrix can be decomposed as:

$$\tilde{K} = K_1 + K_2 + K_3 + \cdots = (M*Q)K_0(M*Q)^T + M(K_0 * K_1)M^T + M(K_0 * K_2)(M^T) + \ldots \quad (0)$$

i.e., $K_1 = (M*Q)K_0(M*Q)^T$ and $K_{i+1} = M(K_0 * K_i)M^T \quad (i = 1, 2, \cdots\infty)$ where * is the Hadamard product (i.e., entrywise product) where $A*B=\{a_{i,j}\cdot b_{i,j}\}$.

Proof:

According to our definition of the context graph kernel:

$$K(G_x, G_y) = \sum_{h_i \in H(G_x)} \sum_{h_j \in H(Gy)} K_h(h_i, h_j) P(h_i | G_x) P(h_j | G_y). \quad (1)$$

Considering that $K_h(h_i, h_j) = 0$ when the lengths of h_i and h_j are not equal, and $K_h(h_i, h_j) = \prod_k K_g(g_{<h_i,k>}, g_{<h_j,k>})$ when the lengths of h_i and h_j are equal, the random walk paths with the same length can be grouped together. The context graph kernel becomes:

$$K(G_x, G_y) = \lim_{L \to \infty} \sum_{l=1}^{L} \left(\sum_{|h_i|=l} \sum_{|h_j|=l} \left(\prod_{k=1}^{l} K_g(g_{<h_i,k>}, g_{<h_j,k>}) \right) P(h_i \mid G_x) P(h_j \mid G_y) \right), \qquad (2)$$

where $|h_i|$ and $|h_j|$ are the lengths of random walk paths h_i and h_j.

For random walk path $h = (g_{<h,0>} \to g_{<h,1>} \to \ldots \to g_{<h,l>})$, the probability to exist is

$$P(h \mid G) = p_t(g_{<h,1>} \mid g_{<h,0>}) p_t(g_{<h,2>} \mid g_{<h,1>}) \cdots p_t(g_{<h,l>} \mid g_{<h,l-1>}) p_s(g_{<h,l>}). \qquad (3)$$

Combining formula (3) into formula (2), the context graph kernel can be represented as:

$$\begin{aligned} K(G_x, G_y) = \lim_{L \to \infty} \sum_{l=1}^{L} \Bigg(\sum_{|h_i|=l} \sum_{|h_j|=l} \prod_{k=1}^{l} \Big(& K_g(g_{<h_i,k>}, g_{<h_j,k>}) p_t(g_{<h_i,k>} \mid g_{<h_i,k-1>}) \\ & p_t(g_{<h_j,k>} \mid g_{<h_j,k-1>}) \Big) p_s(g_{<h_i,l>}) p_s(g_{<h_j,l>}) \Bigg). \end{aligned} \qquad (4)$$

By converting the random walk paths in the pair-wised summation to node sequences and changing the order of summation and multiplication (Kashima et al., 2003, 2004), formula (4) is converted to:

$$
\begin{aligned}
& K(G_x, G_y) = \\
& \lim_{L\to\infty} \sum_{l=1}^{L} \Bigg(\sum_{<h_i,1>,<h_j,1>} K_g(g_{<h_i,1>}, g_{<h_j,1>}) p_t(g_{<h_i,1>} \mid g_{<h_i,0>}) p_t(g_{<h_j,1>} \mid g_{<h_j,0>}) \times \\
& \qquad \Bigg(\sum_{<h_i,2>,<h_j,2>} K_g(g_{<h_i,2>}, g_{<h_j,2>}) p_t(g_{<h_i,2>} \mid g_{<h_i,1>}) p_t(g_{<h_j,2>} \mid g_{<h_j,1>}) \times \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Bigg(\cdots \times \ , \qquad\qquad (5) \\
& \quad \Bigg(\sum_{<h_i,l>,<h_j,l>} K_g(g_{<h_i,l>}, g_{<h_j,l>}) p_t(g_{<h_i,l>} \mid g_{<h_i,l-1>}) p_t(g_{<h_j,l>} \mid g_{<h_j,l-1>}) \times \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad p_s(g_{<h_i,l>}) p_s(g_{<h_j,l>}) \ \Bigg) \\
& \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \cdots \ \Bigg)\Bigg)\Bigg)
\end{aligned}
$$

which can be represented as *K(G_x,G_y)*= $\lim_{L\to\infty}\sum_{l=1}^{L} K(G_x, G_y)_l$. (6)

We now work on proving that $K(G_x,G_y)_l$ in formula (6) is the element of kernel matrix K_l in formula (0):

a) For l=1

Noticing that $x=<h_i,0>$ and $y= <h_j,0>$, from formula (5) we have:

$$K(G_x, G_y)_1 = K(G_{<h_i,0>}, G_{<h_j,0>})_1 \tag{7}$$

$$= \sum_{<h_i,1>,<h_j,1>} K_g(g_{<h_i,1>}, g_{<h_j,1>}) p_t(g_{<h_i,1>} \mid g_{<h_i,0>}) p_t(g_{<h_j,1>} \mid g_{<h_j,0>}) p_s(g_{<h_i,1>}) p_s(g_{<h_j,1>})$$

$$= \sum_{<h_i,1>,<h_j,1>} K_g(g_{<h_i,1>}, g_{<h_j,1>}) p_t(g_{<h_i,1>} \mid g_{<h_i,0>}) p_s(g_{<h_i,1>}) p_t(g_{<h_j,1>} \mid g_{<h_j,0>}) p_s(g_{<h_j,1>})$$

$$= \sum_{<h_i,1>,<h_j,1>} \left(p_t(g_{<h_i,1>} \mid g_{<h_i,0>}) p_s(g_{<h_i,1>}) \right) K_g(g_{<h_i,1>}, g_{<h_j,1>}) \left(p_t(g_{<h_j,1>} \mid g_{<h_j,0>}) p_s(g_{<h_j,1>}) \right).$$

From the provided matrices in the proposition, we have:

$$M * Q = \{M_{i,j} \times Q_{i,j}\} = \{p_t(g_j \mid g_i) p_s(g_j)\}, \tag{8}$$

where * is Hadamard product (i.e., entrywise product) of the two matrices.

Thus, one element in the matrix K_1:

$$(K_1)_{x,y} = \left((M * Q) K_0 (M * Q)^T\right)_{x,y}$$

$$= \sum_{z=1}^{n} \left((M * Q) K_0\right)_{x,z} \left((M * Q)^T\right)_{z,y}$$

$$= \sum_{z=1}^{n} \sum_{z'=1}^{n} \left((M * Q)\right)_{x,z'} \left(K_0\right)_{z',z} \left((M * Q)^T\right)_{z,y}. \tag{9}$$

Since x=*<hi,0>* and y= *<hj,0>*, and $\left((M * Q)^T\right)_{z,y} = \left((M * Q)\right)_{y,z}$,

$$(K_1)_{x,y} \tag{10}$$

$$= \sum_{<h_i,1>,<h_j,1>} \left(p_t(g_{<h_i,1>} \mid g_{<h_i,0>}) p_s(g_{<h_i,1>}) \right) K_g(g_{<h_i,1>}, g_{<h_j,1>}) \left(p_t(g_{<h_j,1>} \mid g_{<h_j,0>}) p_s(g_{<h_j,1>}) \right).$$

Comparing formula (7) with formula (10), we see that $(K_1)_{x,y} = K(G_x, G_y)_1$.

b) For *l*= 2:

From formula (5) we have:

$$K(G_x, G_y)_2 = \sum_{<h_i,1>,<h_j,1>} K_g(g_{<h_i,1>}, g_{<h_j,1>}) p_t(g_{<h_i,1>} \mid g_{<h_i,0>}) p_t(g_{<h_j,1>} \mid g_{<h_j,0>}) \times$$

$$\left(\sum_{<h_i,2>,<h_j,2>} K_g(g_{<h_i,2>}, g_{<h_j,2>}) p_t(g_{<h_i,2>} \mid g_{<h_i,1>}) p_t(g_{<h_j,2>} \mid g_{<h_j,1>}) p_s(g_{<h_i,2>}) p_s(g_{<h_j,2>}) \right)$$

$$= \sum_{<h_i,1>,<h_j,1>} K_g(g_{<h_i,1>}, g_{<h_j,1>}) p_t(g_{<h_i,1>} \mid g_{<h_i,0>}) p_t(g_{<h_j,1>} \mid g_{<h_j,0>}) \times \left(K(G_{<h_i,1>}, G_{<h_j,1>})_1 \right). \quad (11)$$

From the proposition, one element in K2 is:

$$(K_2)_{x,y} = \left(M(K_1 * K_0) M^T \right)_{x,y}$$

$$= \sum_{z=1}^{n} \sum_{z'=1}^{n} \left(M \right)_{x,z'} \left(K_1 * K_0 \right)_{z',z} \left(M^T \right)_{z,y}$$

$$= \sum_{<h_i,1>,<h_j,1>} p_t(g_{<h_i,1>} \mid g_{<h_i,0>}) \left(K_g(g_{<h_i,1>}, g_{<h_j,1>}) \times \left(K(G_{<h_i,1>}, G_{<h_j,1>})_1 \right) \right) p_t(g_{<h_j,1>} \mid g_{<h_j,0>})$$

$$= \sum_{<h_i,1>,<h_j,1>} K_g(g_{<h_i,1>}, g_{<h_j,1>}) p_t(g_{<h_i,1>} \mid g_{<h_i,0>}) p_t(g_{<h_j,1>} \mid g_{<h_j,0>}) \times \left(K(G_{<h_i,1>}, G_{<h_j,1>})_1 \right). \quad (12)$$

Comparing formula (11) with formula (12), we see that $(K_2)_{x,y} = K(G_x, G_y)_2$.

c) Providing any *l*>=2, if $(K_l)_{x,y} = K(G_x, G_y)_l$, we can follow the procedure in (b) and prove that $(K_{l+1})_{x,y} = K(G_x, G_y)_{l+1}$.

From this procedure, we can prove that any $K(G_x,G_y)_l$ in formula (6) is the element of kernel matrix K_l in formula (0). Since formula (6) was deduced from the design of the context graph kernel, we have proved that the kernel matrix can be decomposed to the forms in formula (0).

APPENDIX B: CONVERGENCE OF THE CONTEXT GRAPH KERNEL

Proposition:

For the context graph kernel, if node information $K_g()$ is normalized and 1) there is uniform stopping probability $p_s(g_i)$ or 2) $p_s(g_i)$ >0.5, then the calculation of the kernel converges when more levels of interactions are included in the context graph, i.e., the process of $\tilde{K} = K_1 + K_2 + K_3 + \cdots + K_i$ converges when i approaches ∞ .

Proof:

We first represent each element in the kernel matrix as

$$K(G_x,G_y)= K(G_x,G_y)_1+ K(G_x,G_y)_2+ K(G_x,G_y)_3+\ldots \qquad (1)$$

All similarity measures are non-negative numbers in the kernel matrix. According to the rule of d'alembert (i.e., ratio test), such a non-negative series summation converges if $\lim_{k\to\infty} \frac{K(G_x,G_y)_{k+1}}{K(G_x,G_y)_k} < 1$.

We can represent $K(G_x,G_y)_{\mathrm{k}}$ and $K(G_x,G_y)_{\mathrm{k+1}}$as:

$$\begin{aligned} K(G_x,G_y)_k = & \sum_{<h_i,1>,<h_j,1>} K_g(g_{<h_i,1>}, g_{<h_j,1>})\, p_t(g_{<h_i,1>} \mid g_{<h_i,0>})\, p_t(g_{<h_j,1>} \mid g_{<h_j,0>}) \times \\ & \Bigg(\sum_{<h_i,2>,<h_j,2>} K_g(g_{<h_i,2>}, g_{<h_j,2>})\, p_t(g_{<h_i,2>} \mid g_{<h_i,1>})\, p_t(g_{<h_j,2>} \mid g_{<h_j,1>}) \times \\ & \qquad \Bigg(\cdots \times \\ & \Bigg(\sum_{<h_i,k>,<h_j,k>} K_g(g_{<h_i,k>}, g_{<h_j,k>})\, p_t(g_{<h_i,k>} \mid g_{<h_i,k-1>})\, p_t(g_{<h_j,k>} \mid g_{<h_j,k-1>}) \times \\ & \qquad p_s(g_{<h_i,k>})\, p_s(g_{<h_j,k>}) \Bigg) \cdots \Bigg)\Bigg) \end{aligned} \tag{2}$$

and

$$\begin{aligned} K(G_x,G_y)_{k+1} = & \sum_{<h_i,1>,<h_j,1>} K_g(g_{<h_i,1>}, g_{<h_j,1>})\, p_t(g_{<h_i,1>} \mid g_{<h_i,0>})\, p_t(g_{<h_j,1>} \mid g_{<h_j,0>}) \times \\ & \Bigg(\sum_{<h_i,2>,<h_j,2>} K_g(g_{<h_i,2>}, g_{<h_j,2>})\, p_t(g_{<h_i,2>} \mid g_{<h_i,1>})\, p_t(g_{<h_j,2>} \mid g_{<h_j,1>}) \times \\ & \qquad \Bigg(\cdots \times \\ & \Bigg(\sum_{<h_i,k+1>,<h_j,k+1>} K_g(g_{<h_i,k+1>}, g_{<h_j,k+1>})\, p_t(g_{<h_i,k+1>} \mid g_{<h_i,k>})\, p_t(g_{<h_j,k+1>} \mid g_{<h_j,k>}) \times \\ & \qquad p_s(g_{<h_i,k+1>})\, p_s(g_{<h_j,k+1>}) \Bigg) \cdots \Bigg)\Bigg) \end{aligned} \tag{3}$$

Thus,

$$\frac{K(G_x,G_y)_{k+1}}{K(G_x,G_y)_k} =$$

$$\frac{\displaystyle\sum_{<h_i,k+1>,<h_j,k+1>} K_g(g_{<h_i,k+1>}, g_{<h_j,k+1>})\, p_t(g_{<h_i,k+1>} \mid g_{<h_i,k>})\, p_t(g_{<h_j,k+1>} \mid g_{<h_j,k>})\, p_s(g_{<h,k+1>})\, p_s(g_{<h_j,k+1>})}{p_s(g_{<h,k>})\, p_s(g_{<h_j,k>})} \tag{4}$$

Since node information $K_g()$ is normalized, each of its elements is between 0 and 1, thus:

$$\frac{K(G_x,G_y)_{k+1}}{K(G_x,G_y)_k} \leq$$

$$\frac{\sum\limits_{<h_i,k+1>,<h_j,k+1>} p_t(g_{<h_i,k+1>} \mid g_{<h_i,k>}) p_t(g_{<h_j,k+1>} \mid g_{<h_j,k>}) p_s(g_{<h_i,k+1>}) p_s(g_{<h_j,k+1>})}{p_s(g_{<h_i,k>}) p_s(g_{<h_j,k>})} \tag{5}$$

1) For the first condition, if $p_s(g_i)$ is unified for every gene:

$$\frac{K(G_x,G_y)_{k+1}}{K(G_x,G_y)_k} \leq \sum_{<h_i,k+1>,<h_j,k+1>} p_t(g_{<h_i,k+1>} \mid g_{<h_i,k>}) p_t(g_{<h_j,k+1>} \mid g_{<h_j,k>})$$

$$\leq \left(1 - p_s(g_{<h_i,k>})\right)\left(1 - p_s(g_{<h_j,k>})\right) < 1 \tag{6}$$

2) For the second condition, if $p_s(g_i)$ is larger than 0.5 for every gene:

Since $p_s(g_i)$ <1, we can first relax

$$\frac{K(G_x,G_y)_{k+1}}{K(G_x,G_y)_k} \leq \frac{\sum\limits_{<h_i,k+1>,<h_j,k+1>} p_t(g_{<h_i,k+1>} \mid g_{<h_i,k>}) p_t(g_{<h_j,k+1>} \mid g_{<h_j,k>})}{p_s(g_{<h_i,k>}) p_s(g_{<h_j,k>})}$$

$$\leq \frac{\left(1 - p_s(g_{<h_i,k>})\right)\left(1 - p_s(g_{<h_j,k>})\right)}{p_s(g_{<h_i,k>}) p_s(g_{<h_j,k>})} \tag{7}$$

Since $p_s(g_i)$ >0.5, it is easy to see $\frac{K(G_x,G_y)_{k+1}}{K(G_x,G_y)_k}$ <1 (8)

Thus, for both conditions, we can get $\lim\limits_{k\to\infty} \frac{K(G_x,G_y)_{k+1}}{K(G_x,G_y)_k} < 1$, and the kernel converges when more levels of interactions are included in the context graph.

REFERENCES

Abello, J., P.M. Pardalos, M.G.C. Resende. 1999. On maximum clique problems in very large graphs. In: Abello, J., J. Vitter (Eds.) *External Memory Algorithms: DIMACS Series on Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, pp. 119-130.

Adomavicius, G., A. Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*. 17(6) 734-749.

Ahn, H.J. 2008. A new similarity measure for collaborative filtering to alleviate the new user cold-starting problem. *Information Sciences*. 178(1) 37-51.

Airoldi, E.M., D.M. Blei, S.E. Fienberg, E.P. Xing. 2008. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*. 9 1981-2014.

Albert, R., A.L. Barabasi. 2002. Statistical mechanics of complex networks. *Reviews of Modern Physics*. 74(1) 47-97.

Almeida, P., B. Kogut. 1999. Localization of knowledge and the mobility of engineers in regional networks. *Management Science*. 45(7) 905-917.

Altschul, S.F., T.L. Madden, A.A. Schaffer, J.H. Zhang, Z. Zhang, W. Miller, D.J. Lipman. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*. 25(17) 3389-3402.

Alves, N.A. 2007. Unveiling community structures in weighted networks. *Physical Review E*. 76(3).

Amsler, R. 1972. *Application of Citation-based Automatic Classification*. University of Texas at Austin, Linguistics Research Center, Austin, TX.

Arenas, A., A. Fernandez, S. Fortunato, S. Gomez. 2008a. Motif-based communities in complex networks. *Journal of Physics a-Mathematical and Theoretical*. 41(22).

Arenas, A., A. Fernandez, S. Gomez. 2008b. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*. 10.

Ashburner, M., C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, M.A. Harris, D.P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J.C. Matese, J.E. Richardson, M. Ringwald, G.M. Rubin, G. Sherlock, G.O. Consortium. 2000. Gene Ontology: tool for the unification of biology. *Nature Genetics*. 25(1) 25-29.

Barabasi, A.L., R. Albert. 1999. Emergence of scaling in random networks. *Science*. 286(5439) 509-512.

Barabasi, A.L., Z.N. Oltvai. 2004. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*. 5(2) 101-113.

Barber, M.J. 2007. Modularity and community detection in bipartite networks. *Physical Review E*. 76(6).

Barlas, I., A. Ginart, J.L. Dorrity. 2005. Self-evolution in knowledge bases. *In the IEEE Autotestcon 2005*, 325-331.

Berry, M.W. 2004. *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer.

Bieber, M., D. Engelbart, R. Furuta, S.R. Hiltz, J. Noll, J. Preece, E.A. Stohr, M. Turoff, B. Van de Walle. 2002. Toward virtual community knowledge evolution. *Journal of Management Information Systems*. 18(4) 11-35.

Bilke, S., C. Peterson. 2001. Topological properties of citation and metabolic networks. *Physical Review E*. 64(3 Pt 2) 036106.

Bollen, J., M.L. Nelson, G. Geisler, R. Araujo. 2007. Usage derived recommendations for a video digital library. *Journal of Network and Computer Applications*. 30(3) 1059-1083.

Borgwardt, K.M., C.S. Ong, S. Schonauer, S.V.N. Vishwanathan, A.J. Smola, H.P. Kriegel. 2005. Protein function prediction via graph kernels. *Bioinformatics*. 21 I47-I56.

Breitkreutz, B.J., C. Stark, M. Tyers. 2003. The GRID: The General repository for interaction datasets. *Genome Biology*. 4(3).

Broder, A., R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, J. Wiener. 2000. Graph structure in the Web. *Computer Networks-the International Journal of Computer and Telecommunications Networking*. 33(1-6) 309-320.

Bruque, S., J. Moyano, J. Eisenberg. 2008. Individual adaptation to IT-induced change: The role of social networks. *Journal of Management Information Systems*. 25(3) 177-206.

Calado, P., M. Cristo, M.A. Goncalves, E.S. de Moura, B. Ribeiro-Neto, N. Ziviani. 2006. Link-based similarity measures for the classification of Web documents. *Journal of the American Society for Information Science and Technology*. 57(2) 208-221.

Capocci, A., V.D.P. Servedio, G. Caldarelli, F. Colaiori. 2005. Detecting communities in large networks. *Physica a-Statistical Mechanics and Its Applications*. 352(2-4) 669-676.

Carlisle, J.P. 2006. Escaping the veil of maya: Wisdom and the organization. *In the 39th Annual Hawaii International Conference on System Sciences,*, 162a-162a.

Chakrabarti, A.K., I. Dror, N. Eakabuse. 1993. Interorganizational transfer of knowledge - an analysis of patent citations of a defense firm. *IEEE Transactions on Engineering Management*. 40(1) 91-94.

Chakrabarti, S., B. Dom, P. Indyk. 1998. Enhanced hypertext categorization using hyperlinks. *In The Annual ACM SIGMOD/PODS Conference*, Seattle, WA, 307-318.

Chang, C.-C., C.-J. Lin 2001. LIBSVM: A library for support vector machines (http://www.csie.ntu.edu.tw/~cjlin/libsvm).

Chen, C.M., D. Hicks. 2004. Tracing knowledge diffusion. *Scientometrics*. 59(2) 199-211.

Chen, H., B.M. Sharp. 2004. Content-rich biological network constructed by mining PubMed abstracts. *BMC Bioinformatics*. 5(1) 147.

Chua, C.E.H., J. Wareham, D. Robey. 2007a. The role of online trading communities in managing internet auction fraud. *MIS Quarterly*. 31(4) 759-781.

Chua, H.N., W.K. Sung, L. Wong. 2006. Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*. 22(13) 1623-1630.

Chua, H.N., W.K. Sung, L. Wong. 2007b. Using indirect protein interactions for the prediction of Gene Ontology functions. *BMC Bioinformatics*. 8.

Clauset, A., M.E.J. Newman, C. Moore. 2004. Finding community structure in very large networks. *Physical Review E*. 70(6).

Craven, M., D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, S. Slattery. 2000. Learning to construct knowledge bases from the World Wide Web. *Artificial Intelligence*. 118(1-2) 69-113.

Craven, M., S. Slattery. 2001. Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning*. 43(1-2) 97-119.

Cristianini, N., J. Shawe-Taylor. 2000. *An Introduction to Support Vector Machines (and other kernel-based learning methods)*. Cambridge University Press, Cambridge.

Cristo, M., P. Calado, E.S. de Moura, N. Ziviani, B. Ribeiro-Neto. 2003. Link information as a similarity measure in Web classification. *In the Symposium on String Processing and Information Retrieval*, 43-55.

Danon, L., A. Diaz-Guilera, J. Duch, A. Arenas. 2005. Comparing community structure identification. *Journal of Statistical Mechanics-Theory and Experiment* -.

D'haeseleer, P., S.D. Liang, R. Somogyi. 2000. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*. 16(8) 707-726.

Donetti, L., M.A. Munoz. 2004. Detecting network communities: a new systematic and efficient algorithm. *Journal of Statistical Mechanics-Theory and Experiment* -.

Duch, J., A. Arenas. 2005. Community detection in complex networks using extremal optimization. *Physical Review E*. 72(2).

Enright, A.J., V. Kunin, C.A. Ouzounis. 2003. Protein families and TRIBES in genome sequence space. *Nucleic Acids Research*. 31(15) 4632-4638.

Enright, A.J., S. Van Dongen, C.A. Ouzounis. 2002. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*. 30(7) 1575-1584.

Erdos, P., A. Renyi. 1959. On random graphs. *Publicationes Mathematicae Debrecen*. 6 290-297.

Fall, C.J., A. Torcsvari, K. Benzineb, G. Karetka. 2003. Automated categorization in the International patent classification. *ACM SIGIR Forum*. 37(1) 10-25.

Fall, C.J., A. Torcsvari, P. Fievet, G. Karetka. 2004. Automated categorization of German-language patent documents. *Expert Systems with Applications*. 26(2) 269-277.

Farkas, I.J., D. Abel, G. Palla, T. Vicsek. 2007. Weighted network modules. *New Journal of Physics*. 9.

Fayyad, U.M., G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy. 1996. *Advances in Knowledge Discovery and Data Mining*. AAAI Press/MIT Press, Menlo Park, CA.

Fell, D.A., A. Wagner. 2000. The small world of metabolism. *Nature Biotechnology*. 18(11) 1121-1122.

Ferrer, i., R. Cancho, R.V. Sole. 2001. The small world of human language. *Proceedings of the Royal Society of London- Series B, Biological Sciences*. 268(1482) 2261-2265.

Flake, G.W., S. Lawrence, C.L. Giles, F.M. Coetzee. 2002. Self-organization and identification of web communities. *Computer*. 35(3) 66.

Fleming, L. 2001. Recombinant uncertainty in technological search. *Management Science*. 47(1) 117-132.

Fortunato, S., M. Barthelemy. 2007. Resolution limit in community detection. *Proceedings of the National Academy of Sciences of the United States of America*. 104(1) 36-41.

Fortunato, S., V. Latora, M. Marchiori. 2004. Method to find community structures based on information centrality. *Physical Review E*. 70(5).

Fouss, F., A. Pirotte, J.M. Renders, M. Saerens. 2007. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*. 19(3) 355-369.

Fu, T.J., A. Abbasi, H.C. Chen. 2008. A hybrid approach to web forum interactional coherence analysis. *Journal of the American Society for Information Science and Technology*. 59(8) 1195-1209.

Furnkranz, J. 2002. Hyperlink ensembles: a case study in hypertext classification. *Information Fusion*. 3(4) 299-312.

Gallini, N.T. 2002. The economics of patents: Lessons from recent US patent reform. *Journal of Economic Perspectives*. 16(2) 131-154.

Gartner, T. 2003. A survey of kernels for structured data. *ACM SIGKDD Explorations*. 5(1) 49-58.

Gavin, A.C., M. Bosche, R. Krause, P. Grandi, M. Marzioch, A. Bauer, J. Schultz, J.M. Rick, A.M. Michon, C.M. Cruciat, M. Remor, C. Hofert, M. Schelder, M. Brajenovic, H. Ruffner, A. Merino, K. Klein, M. Hudak, D. Dickson, T. Rudi, V. Gnau, A. Bauch, S. Bastuck, B. Huhse, C. Leutwein, M.A. Heurtier, R.R. Copley, A. Edelmann, E. Querfurth, V. Rybin, G. Drewes, M. Raida, T. Bouwmeester, P. Bork, B. Seraphin, B. Kuster, G. Neubauer, G. Superti-Furga. 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*. 415(6868) 141-147.

Gemino, A., D. Parker, A.O. Kutzschan. 2005. Investigating coherence and multimedia effects of a technology-mediated collaborative environment. *Journal of Management Information Systems*. 22(3) 97-121.

Getoor, L., M. Sahami. 1999. Using probabilistic relational models for collaborative filtering. *In the International WebKDD Workshop*.

Ghanem, M.M., Y. Guo, H. Lodhi, Y. Zhang. 2002. Automatic scientific text classification using local patterns: KDD CUP 2002 (task 1). *ACM SIGKDD Explorations Newsletter*. 4(2) 95-96.

Ghani, R., S. Slattery, Y. Yang. 2001. Hypertext categorization using hyperlink patterns and meta data. *In the 18th International Conference on Machine Learning*, 178-185.

Ginsparg, P., P. Houle, T. Joachims, J.H. Sul. 2004. Mapping subsets of scholarly information. *Proceedings of the National Academy of Sciences of the United States of America*. 101 5236-5240.

Girvan, M., M.E.J. Newman. 2002. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences of the United States of America*. 99(12) 7821-7826.

Gori, M., A. Pucci. 2007. ItemRank: A random-walk based scoring algorithm for recommender engines. *In International Joint Conference on Artificial Intelligence*.

Griffith, J., C. O'Riordan, H. Sorensen. 2006. A constrained spreading activation approach to collaborative filtering. *In the International Conference on Knowledge-Based Intelligent Information and Engineering Systems*, 766-773.

Gu, B., P. Konana. 2007. Competition among virtual communities and user valuation: The case of investing-related communities. *Information Systems Research*. 18(1) 68-85.

Guimera, R., L.A.N. Amaral. 2005. Functional cartography of complex metabolic networks. *Nature*. 433(7028) 895-900.

Guimera, R., L. Danon, A. Diaz-Guilera, E. Giralt, A. Arenas. 2006. The real communication network behind the formal chart: Community structure in organizations. *Journal of Economic Behavior & Organization*. 61(4) 653-667.

Hahn, S., R. Ladner, M. Ostendorf. 2006. Agreement/disagreement classification: Exploiting unlabeled data using contrast classifiers. *In the Human Language Technology Conference of the NAACL*, New York City, USA.

Hasan, M.A., V. Chaoji, S. Salem, M. Zaki. 2006. Link prediction using supervised learning. *In Workshop on Link Analysis, Counter-terrorism and Security*.

Hastings, M.B. 2006. Community detection as an inference problem. *Physical Review E*. 74(3) -.

Haussler, D. 1999. *Convolution kernels on discrete structures*. UC Santa Cruz.

Hevner, A.R., S.T. March, J. Park, S. Ram. 2004. Design science in Information Systems research. *MIS Quarterly*. 28(1) 75-105.

Hishigaki, H., K. Nakai, T. Ono, A. Tanigami, T. Takagi. 2001. Assessment of prediction accuracy of protein function from protein-protein interaction data. *Yeast*. 18(6) 523-531.

Ho, Y., A. Gruhler, A. Heilbut, G.D. Bader, L. Moore, S.L. Adams, A. Millar, P. Taylor, K. Bennett, K. Boutilier, L. Yang, C. Wolting, I. Donaldson, S. Schandorff, J. Shewnarane, M. Vo, J. Taggart, M. Goudreault, B. Muskat, C. Alfarano, D. Dewar, Z. Lin, K. Michalickova, A.R. Willems, H. Sassi, P.A. Nielsen, K.J. Rasmussen, J.R. Andersen, L.E. Johansen, L.H. Hansen, H. Jespersen, A. Podtelejnikov, E. Nielsen, J. Crawford, V. Poulsen, B.D. Sorensen, J. Matthiesen, R.C. Hendrickson, F. Gleeson, T. Pawson, M.F. Moran, D. Durocher, M. Mann, C.W. Hogue, D. Figeys, M. Tyers. 2002. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature*. 415(6868) 180-183.

Hofmann, T. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems*. 22(1) 89-115.

Hopcroft, J., O. Khan, B. Kulis, B. Selman. 2004. Tracking evolving communities in large linked networks. *Proceedings of the National Academy of Sciences of the United States of America*. 101 5249-5253.

Hu, P.Z., G. Bader, D.A. Wigle, A. Emili. 2007. Computational prediction of cancer-gene function. *Nature Reviews Cancer*. 7(1) 23-34.

Huang, M.H., L.Y. Chiang, D.Z. Chen. 2003a. Constructing a patent citation map using bibliographic coupling: A study of Taiwan's high-tech companies. *Scientometrics*. 58(3) 489-506.

Huang, W.W., K.K. Wei. 2000. An empirical investigation of the effects of group support systems (GSS) and task type on group interactions from an influence perspective. *Journal of Management Information Systems*. 17(2) 181-206.

Huang, Z., H. Chen, Z.-K. Chen, M.C. Roco. 2004a. International nanotechnology development in 2003: Country, institution, and technology field analysis based on USPTO patent database. *Journal of Nanoparticale Research*. 6(4) 325-354.

Huang, Z., H. Chen, A. Yip, G. Ng, F. Guo, Z.-K. Chen, M.C. Roco. 2003b. Longitudinal patent analysis for Nanoscale Science and Engineering: Country, institution and technology field. *Journal of Nanoparticale Research*. 5 333-363.

Huang, Z., H. Chen, D. Zeng. 2004b. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems*. 22(1) 116-142.

Huang, Z., W.Y. Chung, H.C. Chen. 2004c. A graph model for e-commerce recommender systems. *Journal of the American Society for Information Science and Technology*. 55(3) 259-274.

Huang, Z., D. Zeng, H. Chen. 2007a. Analyzing consumer-product graphs: Empirical findings and applications in recommender systems. *Management Science*. 53(7) 1146-1164.

Huang, Z., D. Zeng, H. Cheng. 2004d. A unified recommendation framework based on probabilistic relational models. *In the 4th Annual Workshop on Information Technologies and Systems*.

Huang, Z., D. Zeng, H. Cheng. 2007b. A comparative study of recommendation algorithms in e-commerce applications. *IEEE Intelligent Systems*. 22(5) 68-78.

Hull, D., S. Ait-Mokhtar, M. Chuat, A. Eisele, E. Gaussier, G. Grefenstette, P. Isabelle, C. Samuelsson, F. Segond. 2001. Language technologies and patent search and classification. *World Patent Information*. 21(3) 265-268.

Hunt, R.M. 2001. You can patent that? Are patents on computer programs and business methods good for the economy? *Federal Reserve Bank of Philadelphia Business Review*. Q1 5-15.

Huynen, M.A., B. Snel, C. von Mering, P. Bork. 2003. Function prediction and protein networks. *Current Opinion in Cell Biology*. 15(2) 191-198.

Iwata, T., K. Saito, T. Yamada. 2008. Recommendation method for improving customer lifetime value. *IEEE Transactions on Knowledge and Data Engineering*. 20(9) 1254-1263.

Jensen, L.J., R. Gupta, H.H. Staerfeldt, S. Brunak. 2003. Prediction of human protein function according to Gene Ontology categories. *Bioinformatics*. 19(5) 635-642.

Jenssen, T.K., A. Laegreid, J. Komorowski, E. Hovig. 2001. A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*. 28(1) 21-28.

Jeong, H., S.P. Mason, A.L. Barabasi, Z.N. Oltvai. 2001. Lethality and centrality in protein networks. *Nature*. 411(6833) 41-42.

Jeong, H., Z. Neda, A.L. Barabasi. 2003. Measuring preferential attachment in evolving networks. *Europhysics Letters*. 61(4) 567-572.

Jeong, H., B. Tombor, R. Albert, Z.N. Oltvai, A.L. Barabasi. 2000. The large-scale organization of metabolic networks. *Nature*. 407(6804) 651-654.

Joachims, T., N. Cristianini, J. Shawe-Taylor. 2001. Composite kernels for hypertext categorisation. *In the 18th International Conference on Machine Learning*, 250-257.

Johnson, S.C. 1967. Hierarchical clustering schemes. *Psychometrika*. 32(3) 241-241.

Jones, Q., G. Ravid, S. Rafaeli. 2004. Information overload and the message dynamics of online interaction spaces: A theoretical model and empirical exploration. *Information Systems Research*. 15(2) 194-210.

Karaoz, U., T.M. Murali, S. Letovsky, Y. Zheng, C.M. Ding, C.R. Cantor, S. Kasif. 2004. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proceedings of the National Academy of Sciences of the United States of America*. 101(9) 2888-2893.

Karki, M.M. 1997. Patent citation analysis: a policy analysis tool. *World Patent Information*. 19 269-272.

Karypis, G., V. Kumar. 1998. A fast and high quality multilevel scheme for partitioning irregular graphs. *Siam Journal on Scientific Computing*. 20(1) 359-392.

Kashima, H., K. Tsuda, A. Inokuchi. 2003. Marginalized kernels between labeled graphs. *In the 20th International Conference on Machine Learning*.

Kashima, H., K. Tsuda, A. Inokuchi. 2004. Kernels for Graphs. In: *Kernel Methods in Computational Biology*. MIT Press.

Kessler, M.M. 1963. Bibliographic coupling between scientific papers. *American Documentation*. 14(1) 10.

King, J.L. 2003. Patent examination procedures and patent quality. In: Cohen, W.M., S.A. Merrill (Eds.) *Patents in the Knowledge-Based Economy*. National Academies Press, Washington, D.C., pp. 54-73.

Koster, C.H.A., M. Seutter, J. Beney. 2001. Classifying patent applications with winnow. *In the Benelearn 2001 Conference*.

Koster, C.H.A., M. Seutter, J. Beney. 2003. Multi-classification of patent applications with winnow. *Perspectives of System Informatics*. 2890 546-555.

Krier, M., F. Zacca. 2002. Automatic categorisation applications at the European patent office. *World Patent Information*. 24(3) 187-196.

Lanckriet, G.R.G., T. De Bie, N. Cristianini, M.I. Jordan, W.S. Noble. 2004. A statistical framework for genomic data fusion. *Bioinformatics*. 20(16) 2626-2635.

Larkey, L.S. 1999. A patent search and classification system. *In the 4th ACM Conference on Digital Libraries*, Berkeley,CA, 79-87.

Lawrence, S., C.L. Giles. 1998. Searching the World Wide Web. *Science*. 280(5360) 98-100.

Le, S.Q., T.B. Ho, T.T.H. Phan. 2004. A novel graph-based similarity measure for 2D chemical structures. *Genome Informatics*. 14(2) 82-91.

Levy, D.M., C.C. Marshall. 1995. Going digital - A look at assumptions underlying digital libraries. *Communications of the ACM*. 38(4) 77-84.

Li, J., X. Li, H. Su, H.C. Chen, D.W. Galbraith. 2006. A framework of integrating gene relations from heterogeneous data sources: an experiment on Arabidopsis thaliana. *Bioinformatics*. 22(16) 2037-2043.

Li, X., H. Chen, Z. Huang, M.C. Roco. 2007a. Patent citation network in nanotechnology (1976-2004). *Journal of Nanoparticle Research*. 9(3) 337-352.

Li, X., H. Chen, Z. Zhang, J. Li. 2007b. Automatic patent classification using citation network information: an experimental study in nanotechnology. *In the 7th ACM/IEEE Joint Conference on Digital Libraries*, 419-427.

Li, Y.H., Z. Guo, W.C. Ma, D. Yang, D. Wang, M. Zhang, J. Zhu, G.C. Zhong, Y.J. Li, C. Yao, J. Wang. 2007c. Finding finer functions for partially characterized proteins by protein-protein interaction networks. *Chinese Science Bulletin*. 52(24) 3363-3370.

Linden, G., B. Smith, J. York. 2003. Amazon.com recommendation - Item-to-item collaborative filtering. *IEEE Internet Computing*. 7(1) 76-80.

Loh, H.T., C. He, L. Shen. 2006. Automatic classification of patent documents for TRIZ users. *World Patent Information*. 28(1) 6-13.

Luscombe, N.M., M.M. Babu, H. Yu, M. Snyder, S.A. Teichmann, M. Gerstein. 2004. Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature*. 431(7006) 308-312.

Ma, H.W., A.P. Zeng. 2003. The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics*. 19(11) 1423-1430.

Ma, M., R. Agarwal. 2007. Through a glass darkly: Information technology design, identity verification, and knowledge contribution in Online communities. *Information Systems Research*. 18(1) 42-67.

Marshall, B., H. Su, D. McDonald, H. Chen. 2005. Linking ontological resources using aggregatable substance identifiers to organize extracted relations. *In the Pacific Symposium on Biocomputing*, 162-173.

Marshall, B., H. Su, D. McDonald, S. Eggers, H. Chen. 2006. Aggregating automatically extracted regulatory pathway relations. *IEEE Transactions on Information Technology in Biomedicine*. 10(1) 100-108.

Massa, P., P. Avesani. 2007. Trust-aware recommender systems. *In the ACM Recommender Systems Conference*, Minneapolis, Minnesota, USA.

Massjouni, N., C.G. Rivera, T.M. Murali. 2006. VIRGO: computational prediction of gene functions. *Nucleic Acids Research*. 34 W340-W344.

Matheus, C.J., P.K. Chan, G. Piatetskyshapiro. 1993. Systems for knowledge discovery in databases. *IEEE Transactions on Knowledge and Data Engineering*. 5(6) 903-913.

Mayer, M.L., P. Hieter. 2000. Protein networks - built by association. *Nature Biotechnology*. 18(12) 1242-1243.

McCallum, A.K. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering (http://www.cs.cmu.edu/~mccallum/bow).

McDonald, D.M., H. Chen, H. Su, B.B. Marshall. 2004. Extracting gene pathway relations using a hybrid grammar: the Arizona Relation Parser. *Bioinformatics*. 20(18) 3370-3378.

Milgram, S. 1967. Small-world problem. *Psychology Today*. 1(1) 61-67.

Montazemi, A.R., J.J. Siam, A. Esfahanipour. 2008. Effect of network relations on the adoption of electronic trading systems. *Journal of Management Information Systems*. 25(1) 233-266.

Muller, K.R., S. Mika, G. Ratsch, K. Tsuda, B. Scholkopf. 2001. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*. 12(2) 181-201.

Murali, T.M., C.J. Wu, S. Kasif. 2006. The art of gene function prediction. *Nature Biotechnology*. 24(12) 1474-1475.

Nabieva, E., K. Jim, A. Agarwal, B. Chazelle, M. Singh. 2005. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*. 21 I302-I310.

Narin, F. 1994. Patent bibliometrics. *Scientometrics*. 30(1) 147-155.

Nerkar, A. 2003. Old is gold? The value of temporal exploration in the creation of new knowledge. *Management Science*. 49(2) 211-229.

Newman, M.E. 2001. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences of the United States of America*. 98(2) 404-409.

Newman, M.E., M. Girvan. 2004. Finding and evaluating community structure in networks. *Physical Review E*. 69(2 Pt 2) 026113.

Newman, M.E.J. 2004a. Analysis of weighted networks. *Physical Review E*. 70(5).

Newman, M.E.J. 2004b. Fast algorithm for detecting community structure in networks. *Physical Review E*. 69(6) -.

Newman, M.E.J. 2006a. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*. 74(3) -.

Newman, M.E.J. 2006b. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*. 103(23) 8577-8582.

Newman, M.E.J., E.A. Leicht. 2007. Mixture models and exploratory analysis in networks. *Proceedings of the National Academy of Sciences of the United States of America*. 104(23) 9564-9569.

Newton, J., R. Greiner. 2004. Hierarchical probabilistic relational models for collaborative filtering. *In Workshop on Statistical Relational Learning, 21st International Conference on Machine Learning*.

Nidumolu, S.R., M. Subramani, A. Aldrich. 2001. Situated learning and the situated knowledge web: Exploring the ground beneath knowledge management. *Journal of Management Information Systems*. 18(1) 115-150.

Oh, H.-J., S.H. Myaeng, M.-H. Lee. 2000. A practical hypertext categorization method using links and incrementally available class information. *In the 23rd ACM*

International Conference on Research and Development in Information Retrieval, 264-271.

Oh, W., J.N. Choi, K. Kim. 2005. Coauthorship dynamics and knowledge capital: The patterns of cross-disciplinary collaboration in information systems research. *Journal of Management Information Systems*. 22(3) 265-292.

Oppenheim, C. 2000. Do patent citations count? In: Cromin, B., H.B. Atkins (Eds.) *The Web of knowledge*. Information Today, Inc., Medford, pp. 405-432.

Palla, G., A.L. Barabasi, T. Vicsek. 2007. Quantifying social group evolution. *Nature*. 446(7136) 664-667.

Palla, G., I. Derenyi, I. Farkas, T. Vicsek. 2005. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*. 435(7043) 814-818.

Pavlidis, P., J. Weston, J.S. Cai, W.S. Noble. 2002. Learning gene functional classifications from multiple data types. *Journal of Computational Biology*. 9(2) 401-411.

Pazzani, M.J. 1999. A framework for collaborative, content-based and demographic filtering. *Artificial Intelligence Review*. 13(5-6) 393-408.

Piselli, F. 2007. Communities, places, and social networks. *American Behavioral Scientist*. 50(7) 867-878.

Polcicova, G., P. Tino. 2004. Making sense of sparse rating data in collaborative filtering via topographic organization of user preference patterns. *Neural Networks*. 17(8-9) 1183-1199.

Radicchi, F., C. Castellano, F. Cecconi, V. Loreto, D. Parisi. 2004. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences of the United States of America*. 101(9) 2658-2663.

Raghavan, U.N., R. Albert, S. Kumara. 2007. Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E*. 76(3) -.

Reddy, P.K., M. Kitsuregawa, P. Sreekanth, S.S. Rao. 2002. A graph based approach to extract a neighborhood customer community for collaborative filtering. *Databases in Networked Information Systems*. 2544 188-200.

Redner, S. 1998. How popular is your paper? An empirical study of the citation distribution. *European Physical Journal B*. 4(2) 131-134.

Reichardt, J., S. Bornholdt. 2007. Clustering of sparse data via network communities - a prototype study of a large online market. *Journal of Statistical Mechanics-Theory and Experiment*(JUN).

Resnick, P., N. Iacovou, M. Suchak, P. Bergstrom, J. Riedl. 1994. GroupLens: An open architecture for collaborative filtering of netnews. *In the ACM 1994 Conference on Computer Supported Cooperative Work*, 175-186.

Richter, G., A. MacFarlane. 2005. The impact of metadata on the accuracy of automated patent classification. *World Patent Information*. 27(1) 13-26.

Robert, L.P., A.R. Dennis, M.K. Ahuja. 2008. Social capital and knowledge integration in digitally enabled teams. *Information Systems Research*. 19(3) 314-334.

Roddick, J.F., M. Spiliopoulou. 1999. A bibliography of temporal, spatial and spatio-temporal data mining research. *ACM SIGKDD Explorations Newsletter*. 1(1) 34-38.

Rosvall, M., C.T. Bergstrom. 2007. An information-theoretic framework for resolving community structure in complex networks. *Proceedings of the National Academy of Sciences of the United States of America*. 104(18) 7327-7331.

Sarwar, B.M., G. Karypis, J.A. Konstan, J. Reidl. 2001. Item-based collaborative filtering recommendation algorithms. *In the International Conference on the World Wide Web*, 285-295.

Scherer, F.M. 2002. The economics of human gene patents. *Academic Medicine*. 77(12) 1348-1367.

Schlitt, T., K. Palin, J. Rung, S. Dietmann, M. Lappe, E. Ukkonen, A. Brama. 2003. From gene networks to gene function. *Genome Research*. 13(12) 2568-2576.

Scholkopf, B., J.C. Platt, J. Shawe-Taylor, A.J. Smola, R.C. Williamson. 1999. *Estimating the support of a high-dimensional distribution*. Microsoft Research.

Schwikowski, B., P. Uetz, S. Fields. 2000. A network of protein-protein interactions in yeast. *Nature Biotechnology*. 18(12) 1257-1261.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys*. 34(1) 1-47.

Sharan, R., I. Ulitsky, R. Shamir. 2007. Network-based prediction of protein function. *Molecular Systems Biology*. 3.

Shaw, S. 2003. Evidence of scale-free topology and dynamics in gene regulatory networks. *In the 12th International Conference on Intelligent and Adaptive Systems and Software Engineering*, 37-40.

Shen-Orr, S.S., R. Milo, S. Mangan, U. Alon. 2002. Network motifs in the transcriptional regulation network of Escherichia coli. *Nature Genetics*. 31(1) 64-68.

Sia, C.L., B.C.Y. Tan, K.K. Wei. 2002. Group polarization and computer-mediated communication: Effects of communication cues, social presence, and anonymity. *Information Systems Research*. 13(1) 70-90.

Sinclair, G., B. Webber. 2004. Classification from full text: A comparison of canonical sections of scientific papers. *In the 20th International Conference on Computational Linguistics*, 69-72.

Singh, J. 2005. Collaborative networks as determinants of knowledge diffusion patterns. *Management Science*. 51(5) 756-770.

Slattery, S., M. Craven. 1998. Combining statistical and relational methods for learning in hypertext domains. *In the 8th International Conference on Inductive Logic Programming*, 38-52.

Small, H. 1974. Co-citation in scientific literature - New measure of relationship between 2 documents. *Current Contents*(7) 7-10.

Smith, H. 2002. Automation of patent classification. *World Patent Information*. 24(4) 269-271.

Spangler, S., J.T. Kreulen, J. Lessler. 2003. Generating and browsing multiple taxonomies over a document collection. *Journal of Management Information Systems*. 19(4) 191-212.

Sprott, D. 2000. Enterprise resource planning: componentizing the enterprise application packages. *Communications of the ACM*. 43(4) 63-69.

Stapley, B.J., G. Benoit. 2000. Biobibliometrics: information retrieval and visualization from co-occurrences of gene names in Medline abstracts. *Pacific Symposium on Biocomputing* 529-540.

Stavrianou, A., J.-H. Chauchat. 2008. Opinion mining issues and agreement identification in forum texts. *In the Atelier FOuille des Donnees dOPinions (FODOP08)*, Fontainebleau, France.

Tan, Y., J. Wang. 2004. A support vector machine with a hybrid kernel and minimal Vapnik-Chervonenkis dimension. *IEEE Transactions on Knowledge and Data Engineering*. 16(4) 385-395.

Tari, L., C.Baral, P.Dasgupta. 2005. Understanding the global properties of funtionally-related gene networks using the gene ontology. *In the Pacific Symposium on Biocomputing*, 209-220.

Taskar, B., P. Abbeel, D. Koller. 2002. Discriminative probabilistic models of relational data. *In the 18th Conference on Uncertainty in Artificial Intelligence*, 485-492.

Teichert, T., M.-A. Mittermayer. 2002. Text mining for technology mointoring. *In the International Engineering Management Conference*, 596-601.

Tong, A.H., G. Lesage, G.D. Bader, H. Ding, H. Xu, X. Xin, J. Young, G.F. Berriz, R.L. Brost, M. Chang, Y. Chen, X. Cheng, G. Chua, H. Friesen, D.S. Goldberg, J. Haynes, C. Humphries, G. He, S. Hussein, L. Ke, N. Krogan, Z. Li, J.N. Levinson, H. Lu, P. Menard, C. Munyana, A.B. Parsons, O. Ryan, R. Tonikian, T. Roberts, A.M. Sdicu, J. Shapiro, B. Sheikh, B. Suter, S.L. Wong, L.V. Zhang, H. Zhu, C.G. Burd, S. Munro, C. Sander, J. Rine, J. Greenblatt, M. Peter, A. Bretscher, G. Bell, F.P. Roth, G.W. Brown, B. Andrews, H. Bussey, C. Boone. 2004. Global mapping of the yeast genetic interaction network. *Science*. 303(5659) 808-813.

Tsuda, K., W.S. Noble. 2004. Learning kernels from biological networks by maximizing entropy. *Bioinformatics*. 20(Suppl. 1) i326-i333.

Vapnik, V. 1995. *The nature of statistical learning theory*. Springer-Verlag, New York.

Vazquez, A., A. Flammini, A. Maritan, A. Vespignani. 2003. Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*. 21(6) 697-700.

Vinayagam, A., R. Konig, J. Moormann, F. Schubert, R. Eils, K.H. Glatting, S. Suhai. 2004. Applying support vector machines for gene ontology based gene function prediction. *BMC Bioinformatics*. 5.

Vucetic, S., Z. Obradovic. 2005. Collaborative filtering using a regression-based approach. *Knowledge and Information Systems*. 7(1) 1-22.

Walther, J.B. 1996. Computer-mediated communication: Impersonal, interpersonal, and hyperpersonal interaction. *Communication Research*. 23(1) 3-43.

Wasserman, S., K. Faust. 1994. *Social network analysis: methods and applications*. Cambridge University Press.

Watson-Manheim, M.B., F. Belanger. 2007. Communication media repertoires: Dealing with the multiplicity of media choices. *MIS Quarterly*. 31(2) 267-293.

Watts, D.J., S.H. Strogatz. 1998. Collective dynamics of 'small-world' networks. *Nature*. 393(6684) 440-442.

Wellman, B. 2005. Community: From neighborhood to network. *Communications of the ACM*. 48(10) 53-55.

Wilkinson, D.M., B.A. Huberman. 2004. A method for finding communities of related genes. *Proceedings of the National Academy of Sciences of the United States of America*. 101 5241-5248.

Wren, J.D., R. Bekeredjian, J.A. Stewart, R.V. Shohet, H.R. Garner. 2004. Knowledge discovery by automated identification and ranking of implicit relationships. *Bioinformatics*. 20(3) 389-398.

Wu, T.F., C.J. Lin, R.C. Weng. 2004. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research*. 5 975-1005.

Wuchty, S., Z.N. Oltvai, A.L. Barabasi. 2003. Evolutionary conservation of motif constituents in the yeast protein interaction network. *Nature Genetics*. 35(2) 176-179.

Xu, J.A., K. Araki. 2006. A SVM-based personal recommendation system for TV programs. *In the 12th International Multi-Media Modelling Conference*.

Xu, J.Z., Y.J. Li. 2006. Discovering disease-genes by topological features in human protein-protein interaction network. *Bioinformatics*. 22(22) 2800-2805.

Yajima, Y. 2006. One-class support vector machines for recommendation tasks. *the Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. 3918 230-239.

Yamanishi, Y., J.-P. Vert, M. Kanehisa. 2004. Protein network inference from multiple genomic data: a supervised approach. *Bioinformatics*. 20(Suppl. 1) i363-i370.

Yang, Y. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval*. 1 69-90.

Yang, Y.M., S. Slattery, R. Ghani. 2002. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*. 18(2-3) 219-241.

Yook, S.H., Z.N. Oltvai, A.L. Barabasi. 2004. Functional and topological characterization of protein interaction networks. *Proteomics*. 4(4) 928-942.

Yu, K., A. Schwaighofer, V. Tresp, X.W. Xu, H.P. Kriegel. 2004. Probabilistic memory-based collaborative filtering. *IEEE Transactions on Knowledge and Data Engineering*. 16(1) 56-69.

Zeng, C., C.X. Xing, L.Z. Zhou, X.H. Zheng. 2004. Similarity measure and instance selection for collaborative filtering. *International Journal of Electronic Commerce*. 8(4) 115-129.

Zhang, Q.Y., R.S. Segall. 2008. Web mining: A survey of current research, techniques, and software. *International Journal of Information Technology & Decision Making*. 7(4) 683-720.

Zhao, X.M., Y. Wang, L.N. Chen, K. Aihara. 2008. Gene function prediction using labeled and unlabeled data. *BMC Bioinformatics*. 9.

Zhou, H.J. 2003a. Distance, dissimilarity index, and network community structure. *Physical Review E*. 67(6).

Zhou, H.J. 2003b. Network landscape from a Brownian particle's perspective. *Physical Review E*. 67(4).

Zhou, T., J. Ren, M. Medo, Y.C. Zhang. 2007. Bipartite network projection and personal recommendation. *Physical Review E*. 76(4).

Ziegler, C.-N., S.M. McNee, J.A. Konstan, G. Lausen. 2005. Improving recommendation lists through topic diversification. *In the International Conference on the World Wide Web*, 22-32.